**INSIDER THREAT DETECTION ON THE WINDOWS OPERATING SYSTEM
USING VIRTUAL MACHINE INTROSPECTION**

THESIS

Martin H. Crawford

AFIT/GCO/ENG/12-15

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCO/ENG/12-15

INSIDER THREAT DETECTION ON THE WINDOWS OPERATING SYSTEM USING VIRTUAL MACHINE INTROSPECTION

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

Martin H. Crawford, BS

Civ, USAF

June 2012

AFIT/GCO/ENG/12-15

INSIDER THREAT DETECTION ON THE WINDOWS OPERATING SYSTEM
USING VIRTUAL MACHINE INTROSPECTION

Martin H. Crawford, BS
Civ, USAF

Approved:
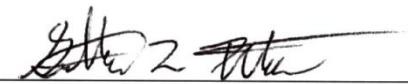
_____
Gilbert L. Peterson, PhD (Chairman)

22 May 2012
Date

_____
Michael R. Grimaila, PhD (Member)

22 Mar 2012
Date
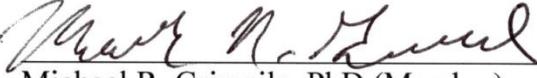
_____
Robert F. Mills, PhD (Member)

22 May 2012
Date

AFIT/GCO/ENG/12-15

Abstract

Existing insider threat defensive technologies focus on monitoring network traffic or events generated by activities on a user's workstation. This research develops a methodology for signaling potentially malicious insider behavior using virtual machine introspection (VMI). VMI provides a novel means to detect potential malicious insiders because the introspection tools remain transparent and inaccessible to the guest and are extremely difficult to subvert. This research develops a four step methodology for development and validation of malicious insider threat alerting using VMI. Six core use cases are developed along with eighteen supporting scenarios. A malicious attacker taxonomy is used to decompose each scenario to aid identification of observables for monitoring for potentially malicious actions. The effectiveness of the identified observables is validated through the use of two data sets, one containing simulated normal and malicious insider user behavior and the second from a computer network operations exercise. Compiled Memory Analysis Tool – Virtual (CMAT-V) and Xen hypervisor capabilities are leveraged to perform VMI and insider threat detection. Results of the research show the developed methodology is effective in detecting all defined malicious insider scenarios used in this research on Windows guests.

Acknowledgments

Table of Contents

List of Figures

xi

List of Tables

INSIDER THREAT DETECTION ON THE WINDOWS OPERATING SYSTEM USING VIRTUAL MACHINE INTROSPECTION

I. Introduction

Users interact with computer workstations to access and manipulate data that is fundamental to the functionality of an organization. As a result, these components of a network contain sensitive information valuable to an attacker, either internal or external. An insider attack against a workstation often results in a breach of the confidentiality, integrity, and/or availability (CIA) [11]. Successful modification of the CIA impacts the organization through data loss, data manipulation, destruction of information, and denial of access to data or a service, all of which negatively impact an organization's efficiency, profit, public image, and overall mission [89].

Unlike external attackers, insiders are already trusted with information on workstations and access portions of it daily. Their trusted position within an organization enables them to cause greater damage. Therefore, developing a monitoring capability to alert for potential insider threats on a workstation can greatly improve defensive potential. Although insider threat monitoring technologies currently exist [4] [77-78] [86], they run at the same privilege level as the insider, allowing the possibility of subversion or determining its capabilities. As such, a monitoring capability which is invisible to a user would be an improvement to mitigating a determined malicious insider.

1.1 Background

Insiders are typically defined as individuals who have association with an organization, have or had access or knowledge of the organization's information systems,

data, policies, and procedures. [4] [6-11]. This definition is used to define the *malicious insider*; a subset of individuals who abuse their trusted position within the organization to accomplish an objective that is not aligned with the organization's mission. This trusted position within a network allows insiders to have the potential to cause much more damage than an external attacker. While financial damage may be the most easily observed, malicious insiders can also cause significant damage to an organization's reputation or a government's alliances with other countries.

Detecting malicious insiders is often difficult due to the trusted nature of their position within the organization. Legitimate user commands or functions, such as printing, removable media, or document access, can easily be leveraged against an organization by an insider.

Within the Department of Defense (DoD) computer networks, it is estimated that approximately eighty-seven percent of intrusions are the result of insider threat actions [19]. This figure illustrates that current mitigation techniques for internal attackers are ineffective compared with the technologies employed to prevent external attackers. External attackers are mitigated using defense-in-depth, which employs multiple tools at various components within a network to require an external attacker to bypass all of them to execute a successful attack.

1.2 Research Objective

Security tools currently exist for monitoring a user's workstation for insider threat actions, but these tools execute at the same privilege level as the user [4] [77-78] [86]. It is believed that through virtual machine introspection, insider threats can be reliably

alerted on a workstation. The primary objective of this research is to develop a methodology to generate alerts for potential malicious insider threat actions and to rely exclusively on virtual machine introspection capabilities of the Compiled Memory Analysis Tool – Virtual (CMAT-V) [47], therefore remaining transparent to potential insiders by executing at a higher privilege level.

Security tools that do address insider threats, such as host based security systems (HBSS) [87] or intrusion detection systems (IDS) [88] are vulnerable to subversion by a malicious insider. HBSS systems running on a user's workstation could be disabled either due to misconfiguration, privilege escalation, or by a user with administrative credentials. Once the workstation defenses are disabled, an organization has no way to monitor the current actions a user is performing, save for a coworker looking over the insider's shoulder. A malicious insider who has root permission, or who collaborates with an external third party could disable or modify current host based monitoring capabilities leaving an organization unaware of actions currently being performed by an insider.

Network layer defenses can be subverted through encryption of traffic, or by avoiding the network layer entirely. An insider employing encryption such as Secure Sockets Layer (SSL), Secure Shell (SSH), or Virtual Private Network (VPN) would defeat any traffic inspection capabilities. A security analyst would only see traffic originating from a user's workstation with the destination at a remote server, they could not determine the contents of the traffic.

A monitoring tool executing at a higher privilege level than an insider could obtain ensures an organization can maintain observation over a user's behavior. Additionally, by running completely transparently to the user, detection of the existence

3

of the tool is extremely difficult and as a result, a potential insider may be more reckless in their attack and not try to conceal their actions. Furthermore, virtual machine introspection has proven successful for detection of malware and post-incident forensics. Therefore, development of alert generation capabilities for malicious insider actions seems plausible and is a logical approach to defending against them.

1.3 Methodology

To alert to a potential malicious insider threat, organizations must develop use cases which categorize possible attack techniques, such as data exfiltration via printing. From a generic use case, specific attack scenarios are developed to enumerate steps a malicious insider may perform.

The taxonomy developed by Howard and Longstaff [50] for a network attacker is modified to be specific to insider threats. Each generated scenario is broken down using this taxonomy to provide a better understanding of the attack. After each action in an attack is identified, corresponding observables are recorded which enable alerting when a specific action is performed.

Once observables for each action are identified, they are tested against malicious insider threat data to confirm the alerting technique for each action is successful. An alert is generated if a potentially malicious action is detected for any observable during a scenario. The alert generation techniques are also compared against two data sets not containing an insider threat. This enables confirmation that the detection techniques only alert for malicious activity and not normal user actions.

1.4 Assumptions and Limitations

The research is limited in terms of accuracy of malicious insider scenarios. Generated scenarios are intended to be representative of realistic attacks performed by insiders, but finding specific details about the techniques insiders used is often difficult. Technical reports often only provide recommendations for organizations to implement and actual incidents, such as [26], often only provide one small piece of information regarding the insider's methods.

Additional limitations are encountered in an effort to generate additional observables for malicious actions. These limitations are accounted for during experiment and analysis portions of the research. Specific details regarding these assumptions can be found in Chapter 3.

1.5 Implications

This research presents a novel method for alerting on potential insider threats. Leveraging VMI enables the alerting method to remain invisible to the individual being monitored. Furthermore, using VMI ensures a malicious insider needs develop a zero day exploit to escape the virtual machine to disable the monitoring capabilities, a difficult task. As previously mentioned, this research is able to alleviate some of the difficulties encountered with current mitigation techniques being defeated by malicious insiders. Additionally, this research provides a reproducible methodology to detect additional insider attack vectors specific to an organization. Six use cases are used to generate eighteen malicious insider attack scenarios. The alert generation techniques developed through taxonomy development and identification of VMI observables successfully

identifies all eighteen malicious insider scenarios. In the non-malicious scenarios, malicious insider action is not detected. Within the Advanced Cyber Education (ACE) Hackfest data set, fifteen scenarios do not have malicious insider activity detected. Three clipboard scenarios could not be determined due to clipboard limitations.

1.6 Thesis Overview

This chapter presented an introduction to the problem of malicious insider threats and the motivation for the research. Chapter 2 provides background information addressing such as: insider threat definition, insider threat impact, insider threat characteristics, insider threat case studies, virtualization of information technology systems, leveraging virtualization to introspect virtual machines, insider threat taxonomies, and insider threat countermeasures. Chapter 3 describes the research approach, the two networks used to collect data, and experimental limitations. Chapter 4 presents each use case, scenarios performed in support of each use case, decomposition of each scenario using a modified computer and network incident taxonomy developed by [50], and virtual machine introspection (VMI) observables. Chapter 5 presents the analysis of the previously identified observables against malicious insider threat data and non-malicious data and explains resulting alerts generated. Chapter 6 summarizes the results, lists possible future work areas, presents information learned from reverse engineering several Windows internal components, and provides conclusions.

II. Literature Review

As discussed in Chapter 1, malicious insiders are much better positioned than an external attacker to cause significant damage to an organization. Malicious insiders are trusted by an organization and conduct their daily job functions, as well as their malicious actions, behind a majority of the organization's network defenses. As a result, additional defensive mechanisms need to be developed and implemented to mitigate this trusted threat.

This chapter describes the background for the research. First, malicious insider characteristics, impact, and attributes are explored. The psychological aspects of malicious insiders and cyber espionage are examined. Virtual machines and virtual machine introspection technologies are discussed. Finally, existing insider threat detection and mitigation methods are examined.

2.1 Malicious Insiders

Investigating the characteristics, impact and attributes of malicious insiders allows for a better understanding of their potential impact to an organization and increased accuracy and usefulness of defense mechanisms. The Defense Security Service (DSS) reports since 1950, twice as many insiders volunteered than were recruited. Additionally, eighty-five percent of those committing espionage were able to successfully transfer information before being caught [5].

2.1.1 Defining the Insider Threat

Insiders are frequently [4] [6-11] defined as individuals who are current or former members of an organization, contractor or partner, who are trusted and have or had access

or knowledge of the organization's information systems, and objectives. *Malicious insiders* is a subset of individuals who intentionally misuse their trusted position through a set of actions and against a target or targets which results in a violation of confidentiality, integrity and/or availability (CIA). Malicious insiders may be disgruntled employees, employees who see an opportunity for financial benefit or spies who join an organization in order to commit espionage or financial fraud. Expanding upon this, malicious insiders within the government sector may also be viewed as traitors or spies [10] [11] [13].

2.1.2 Insider Threat Impact

Insider activities resulting in a breach of CIA can be the result of deliberate malicious activity, inappropriate but not malicious activity or accidental; malicious activities accounted for ninety-three percent of breaches caused by insiders [7]. Malicious insiders accounted for 17 percent of all breaches in 2010, a decrease from previous years. [7] attributes this not to a decrease in malicious insiders, but an increase in external attackers. Malicious insider activities are also under reported by organizations. Organizations attempt to remediate malicious insiders internally for several reasons, such as avoiding personnel problems and potentially very damaging negative publicity [12] [13]. Publically revealing an organization has been a victim of an insider threat would be very damaging to its reputation. Since some insider incidents are quickly dispatched within an organization, the extent of the insider threat cannot be determined.

Quantifying the damage a malicious insider can cause is extremely difficult, especially if the insider's goal is not financial gain. Underreporting, handling incidents internally and insiders who are never caught result potentially inaccurate measurements

of their impact. Within the government sector, a CERT study reported forty-two percent of cases in which insiders were caught a substantial portion caused greater than fifty thousand dollars worth of financial damage to their organization [15]. Within the critical infrastructure sector, thirty-one percent of cases had financial damages greater than fifty thousand dollars [17]. In the financial sector, thirty percent of cases had damages resulting in more than five hundred thousand dollars [14].

Not all malicious insiders' activities can be quantified with a monetary value. Some of the most damaging insider threat events occur against a nation state and are difficult, if not impossible, to apply a monetary value to. In addition to the dissemination of sensitive information, relationships with allies can be damaged. Malicious insiders within the Intelligence Community(IC) are typically performed by spies and not a result of computer network exploitation (CNE) [12]. It is estimated that eighty-seven percent of identified intrusions into Department of Defense (DoD) information systems are a result of insider threat activities [19]. A malicious insider within the IC could provide manipulated data to decision makers resulting in extremely damaging decisions begin made regarding policy towards another country [18].

In addition to violating confidentiality, integrity, and/or availability during employment, insiders also take information with them when they leave an organization. In a survey by the Ponemon Institute focusing on data leaks from 945 respondents who were fired, changed jobs, or laid-off in the past twelve months, seventy-nine percent of respondents took information without company permission. The primary reasons respondents chose to justify this behavior were other laid-off employees had done it, the information may be useful to the employee, and no one checked their property when they

left on their last day. Sixty-seven percent of respondents used information from their previous employer to assist in obtaining new employment. This type of data leak cannot easily be monetarily quantified unless the insiders sold the data and volunteered how much they received for it. Although not directly asked in the survey, it may be inferred that a majority, if not all, of the 945 respondents were not caught because eighty-eight percent stated they had some form of current employment or were a student and the remaining percentage responded as retired, disabled, or other. If these individuals had been caught, it would be much more difficult for them to obtain new employment. This survey illustrates the necessity of observing users during employment to detect malicious insiders earlier and prevent the loss of intellectual property when employees are dismissed or seek other employment opportunities [20].

2.1.3 Insider Threat Characteristics

Malicious insiders do not share a common set of characteristics, technical experience or job position [14-17]. The characteristics of malicious insiders vary based on job sector. Within the government, banking, finance, and critical infrastructure sectors, insiders were approximately fifty percent male; however in the information technology (IT) and telecommunications (Telecom) sectors, ninety-one percent of insiders were male. The majority of male insiders in the IT and Telecom sectors is attributed to the field being primarily male employees [16]. In the IT and Telecom sectors, female employees are as likely to commit insider activities as in other fields; they just represent a much smaller percentage of the workforce than in other fields. The age of malicious insiders is also diverse, with employees ranging from approximately eighteen to sixty years. Based on the characteristics of past insiders, no specific set of characteristics can

be used to generate a profile for potential malicious individuals. Gender, age technical ability cannot be used as indicators to identify individuals who will become malicious in the future or are already malicious [10] [14-17] [20].

Furthermore, malicious insiders did not have a common amount of technical knowledge and were employed in various positions within an organization. Within the banking and financial, and government sectors, twenty-three and twenty-six percent of insiders held a technical position, respectively. A major finding within the CERT study is the lack of technical sophistication needed to perform the malicious activities. Within the government sector, eighty percent of insider actions involved only user commands. Similarly, insiders in banking and financial sectors used regular user commands for eighty-seven percent of actions. However, technical actions used for malicious activities should not be ignored, especially in fields where individuals with technical knowledge compromise a majority of the workforce. Within the IT and Telecom sectors, fifty-eight percent used technical methods such as scripts, tools or backdoors to perform malicious actions [14-16]. Verizon's 2011 Data Breach Investigations Report supports this data with eighty-five percent of insiders being regular employees and nine percent holding technical positions (helpdesk, system or network administrator, or software developer) [7]. Figure 2.1 illustrates how technical sophistication required for executing a cyber attack is decreasing, increasing the ability for malicious insiders to attack their organization. The technical ability required by a malicious insider is decreasing as tools and scripts become more powerful and easier to use. As a result, few insiders need advanced technical knowledge in order to execute their plan [21].
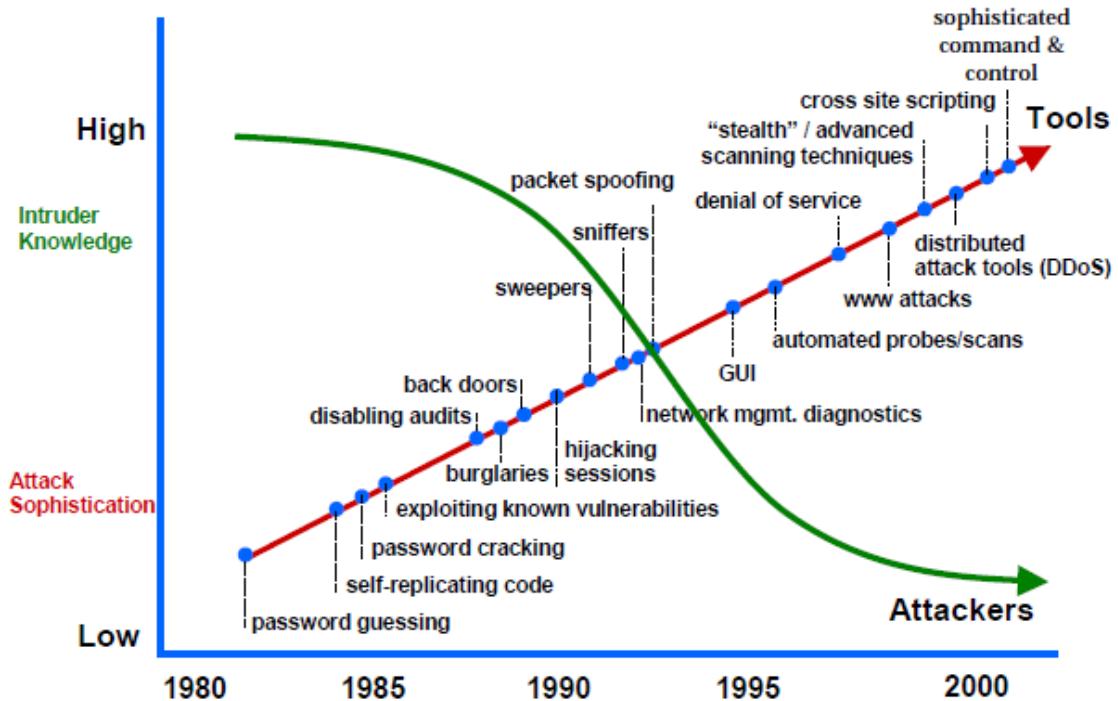
Figure 2.1: Attack Sophistication vs. Intruder Technical Knowledge [21].

The CERT studies on insider threat also reveal the motivation for the insiders to commit their malicious action. In the four studies, the insiders were motivated by a specific event more than fifty percent of the time, with ninety-two percent of critical infrastructure insiders having a specific event triggering their actions. Revenge and financial gain constituted the majority of motivations for malicious insiders. Insider's motives were also based on their employment sector. Employees within the banking and financial sectors were motivated in eighty-one percent of cases by financial gain, whereas employees within the IT and Telecom sectors were motivated primarily by revenge (fifty-six percent of incidents). Some of the less frequent motivations include: lack of appreciation, disagreements with management, culture or policies, and take information with them to a new organization [14-17].

12

Additionally, insiders typically planned their actions before committing their malicious behavior. In three of the four CERT studies, more than three quarters of insiders formulated a plan prior to carrying out their malicious actions. In approximately one third of incidents, malicious insiders performed preliminary actions such as discussing their plan with others, obtaining programs, commands or scripts, testing or attempting to access the system, sabotaging backups, or creating a backdoor. In a majority of cases, insider's plans and negative feelings towards the organization were communicated to others [14-17].

2.2 Insider Threat Detection

To understand the techniques malicious insiders employ in their attacks, it is important to decompose the attack into individual pieces. Taxonomies, or attack trees/models, provide a method for analyzing past and future network attacks.

2.2.1 Attack Models

Computer network attack taxonomies are a standard, formal method for modeling the security of a system based on possible attacks. It enables network defenders to consistently classify network attacks. These taxonomies should also take into account all parts of an attack, such as who is the attacker, in this case malicious insiders, techniques used in attacking, and the targets [48]. A quantified and visual representation of known attack methods allows research efforts to address potential shortfalls of network defense technologies. Figure 2.2 illustrates an example attack taxonomy for insider threats, expanding upon interactions insider threats have with IT systems that can be monitored [49].
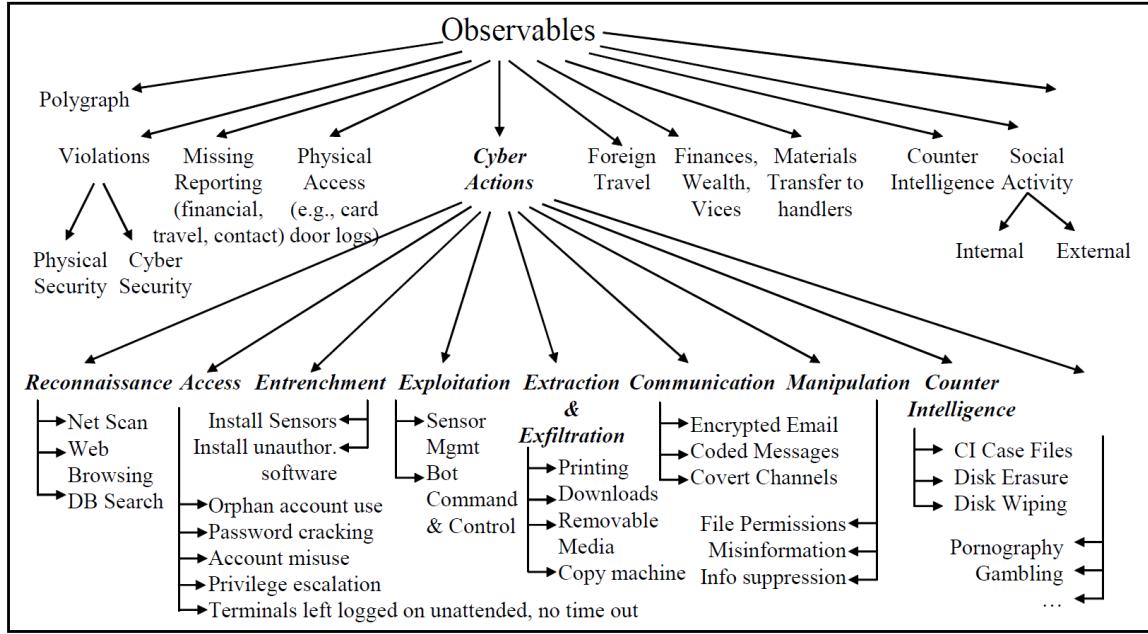
Figure 2.2: Cyber Event/Observable Taxonomy [11].

Traditional attack taxonomies focus on addressing external attacks; that is someone who is external to an organization and often does not have knowledge of the internal network. This research will use a modified version of [50]'s taxonomy to model malicious insider threats. Figure 2.3 presents Howard and Longstaff's Computer and Network Incident Taxonomy. The modified model focuses solely on the insider threat and does not address the motivation for the attack, such as financial gain or revenge. In respect to insider threats, the tool and vulnerability used by a user are extremely important, especially in regards to detection within a Windows OS. Other attack taxonomies, such as [79], are considered, but [50]'s taxonomy allows for rapid identification of multiple portions of an incident. An insider attack is comprised of more than just an action against a target; it is important for an organization to also be able to identify the tools employed, the vulnerability exploited, as well as the insider's objective, and effect of the attack.

14

Attackers	Tool	Vulnerability	Action	Target	Unauthorized Result	Objectives
Hackers	Physical Attack	Design	Probe	Account	Increased Access	Challenge, Status, Thrill
Spies	Information Exchange	Implementation	Scan	Process	Disclosure of Information	Political Gain
Terrorists	User Command	Configuration	Flood	Data	Corruption of Information	Financial Gain
Corporate Raiders	Script or Program		Authenticate	Component	Denial of Service	Damage
Professional Criminals	Autonomous Agent		Bypass	Computer	Theft of Resources	
Vandals	Toolkit		Spoof	Network		
Voyeurs	Distributed Tool		Read	Internetwork		
	Data Tap		Copy			
			Steal			
			Modify			
			Delete			

Figure 2.3: Computer and Network Incident Taxonomy [50].

2.2.2 Honeypots

Honeypots, compared to network technologies such as firewalls or IDSs, are a relatively new technology. A honeypot is a security technology that differs from other network technologies because its function is to be compromised. Honeypots are computers deployed on a network in order to gather information about attacker's tools and techniques. If a honeypot is not attacked, it provides no value for network defenders. Honeypots are useful for detecting "0-day" exploits; that is, an exploit for a previously unknown vulnerability in an OS or service running within the OS. Table 2.1 summarizes the features of honeypots that differentiate them from typical network defenses [51-53]. Honeypots can be developed to imitate an existing service and OS within a network, or

15

run an actual OS. They are typically implemented to be attacked by external agents, sometimes sitting outside the perimeter defenses of a network [3].

Table 2.1: Spitzer Honeypot Benefits [52].

Feature	Description
Small Data Sets	Honeypots only capture data when they are interacted with. This results in less, but more useful data.
Reduced False Positives	False positives are reduced because honeypots provide no services to end-users; therefore any interaction is malicious.
Catching False Negatives	Similar to reduced false positives, any interaction is malicious and this allows honeypots to detect novel attacks.
Encryption	Some network traffic monitors are unable to decrypt encrypted traffic, whereas a honeypot decrypts any encrypted traffic it receives.
IPv6	Although not widely implemented, IPv6 is still not implemented in older devices within networks.
Highly Flexible	Honeypots can be configured to simulate any piece of hardware and contain any type of data an attacker may be looking for.
Minimal Resources	Since honeypots do not provide any network services, they do not require extensive resources like other network defense technologies.

A honeypot is a viable solution or resource to combat the insider threat, when appropriately placed on a network [3]. Since an insider's first step is typically not to attack another computer within the network, honeytokens, or a digital entry, is placed somewhere in a system. It can be any type of information, but its purpose is similar to that of a honeypot [54]. One example demonstrated in [52] is a fake username and password. The login credentials are not used by any user, and are not used on any system. When an insider sniffs the traffic and tries to use the credentials, the organization will know the user is performing malicious actions. Honeytokens could also be used to monitor an employee's data access behavior. Tokens are created that imitate legitimate company data and if an employee accesses the data, it can be monitored for malicious

activity. Furthermore, the insider can be directed to a honeynet, a network of honeypot computers, to determine if the user has malicious intentions [52] [54]. Current efforts [2] seek to include forensic capabilities as part of the honeypot in order to rapidly extract useful information regarding the compromise of the honeypot.

Honeypots are not without their drawbacks with respect to detecting malicious insiders. First, the insider may not use the honeypot while performing malicious actions against an organization, rendering the honeypot insignificant. Secondly, if a honeypot is misconfigured or poorly designed, it may be discovered by an insider and then false information is provided to deceive administrators [52].

2.2.3 Perimeter Network Defenses

Computer network defense techniques often rely on the concept of defense in depth. Specifically focusing on technology, this strategy relies on the idea that an adversary will need to overcome multiple barriers in order to gain access to a specific target [83]. These technologies typically consist of a hardware firewall, an intrusion detection and/or prevention system, and a proxy. In relation to insider threats, these systems attempt to stop sensitive or classified information from being exfiltrated to an external host using methods such as Secure Shell (SSH), Secure Sockets Layer (SSL), Peer to Peer (P2P), or email. A current research effort with network based defense to mitigate the insider threat is Sensitive Information Dissemination Detection (SIDD). SIDD uses deep packet inspection (DPI) in order to retrieve data from within a network packet. It uses anomaly detection to determine if outbound communications are potentially malicious. The system then inspects packets for authorized or unauthorized application traffic. Next, the system attempts to identify the packet contents for any

17

sensitive information using signatures. Lastly, a packet is examined for covert channel communication. SIDD may intervene if it determines network traffic to contain sensitive information [83].

2.2.4 Userlevel/Workstation Defenses

In addition to network based defense, another common method used to mitigate insider threats is monitoring the user's workstation. Compared with network defenses, this is extremely advantageous as monitoring is not limited to only activities that require internet access, but instead research efforts can monitor every aspect of a user's interaction with a computer.

2.2.4.1 Linux Operating System Auditing

The Linux operating system is not as frequently employed as a workstation than its Windows counterpart. As a result, insider threat detection methods are often substantially behind. Current research efforts, such as [77], seek to present a methodology an organization to employ to obtain better information for existing logging functionality of the system. A solution such as [77] requires minimal additional cost and little overhead to employ. This research sought to maximize the logging capabilities of Linux in order to detect insider threats earlier in their attack, rather than after the insider accomplishes their objective.

2.2.4.2 Windows Operating System Logging

Similarly, research by [78] developed a methodology to generate a custom auditing template for the Windows XP OS. Existing Windows logging capabilities are often employed without knowledge of what the organization's actual logging requirements are. A methodology is developed which allows an organization to create an

insider threat logging template tailored to their requirements, thus improving the response time to detect insider threat actions.

2.2.4.3 Physical Hardware Logging

An overlooked area for insider threat detection is logging on physical hardware. Research by [80] presented a methodology and solution to detect insider threat attacks against Cisco network devices. The solution relies only upon existing functionality of the device, meaning implementation is straightforward by an organization and it does not require any additional firmware to be installed on the device for detection to be successful. Network infrastructure devices are often overlooked by security personnel as they are often thought of as not possessing enough storage capability to hold information, but they process all traffic on a network [80].

2.2.2.4 Registry Forensics

The Windows registry is a hierarchical database that stores configuration information about the system, such as configuration information for users, applications and, hardware devices [58]. Table 2.2 describes each of the root keys of the Windows registry.

Table 2.2: Windows Registry Root Keys [42] [58].

Hive Name	Acronym	Function
HKEY_CLASSES_ROOT	HKCR	Contains information correlating a file type to the application which opens it.
HKEY_CURRENT_USER	HKCU	Active, loaded profile for currently logged-on user. Contains information such as Control Panel settings.
HKEY_LOCAL_MACHINE	HKLM	Configuration about the system, including hardware and software.
HKEY_USERS	HKU	Contains all the actively loaded user profiles on the system.
HKEY_CURRENT_CONFIG	HKCC	Contains the system hardware profile used at system startup.

The registry is extremely useful for post-incident forensic analysis of a system. Malware can modify the auto-run features of the registry to start itself after a reboot of the system and forensic investigators often start with the registry to determine if malware has compromised the system. After finding malware in the registry, investigators can determine how the system was compromised [40]. If malware is not found using the registry, investigators can use the registry to determine what software programs are installed on a system and if any are unauthorized. Additionally, the registry can contain valuable forensic information, such as Network Interface Cards (NICs), Media Access Control (MAC) address, network shares, auto run functionality, and removable media [42].

Through thorough examination of the Windows registry, a detailed profile of the user's activities on a computer system can be compiled. Current research efforts, such as [4], present solutions to insider threats through live monitoring of the Windows registry. This technique allows an organization to build a strong profile of a user's computer usage pattern, enabling rapid insider threat detection and mitigation.

2.3 Case Studies

Examination of insider threat case studies expands upon knowledge gained from the previous section. Case studies allow observation of more specific insider threat instances and obtaining information such as vulnerability exploited, tools employed, actions performed, objective of the attack, and consequence of the insider's attack for the organization.

2.3.1 Terry Childs

Terry Childs was a developer and sole network administrator for the city of San Francisco's FiberWAN network. FiberWAN network handled sixty percent of the municipal government's traffic. In June, 2008, he believed he was going to be fired or transferred and refused to give the passwords for the routes and switches to his supervisors. Additionally, he had installed a backdoor in order to maintain access to the system. When the city fired him, Childs demanded three million dollars to cover attorney fees and lost pay [22-24].

In this case, Childs was able to cause an estimated $200,000 of damage to the city as a result of having to pay for consultants to investigate damage to FiberWAN before he divulged the password. Mr. Childs was successful in his attack because he was the only one with knowledge of the FiberWAN infrastructure and passwords. A lack of proper management and failure to adhere to established policies was deemed the source of his ability to inflict damage on the city. Unfortunately in situations where there is no monitoring, automated or manual, insiders are much more successful in their malicious actions [24-25].

2.3.2 Bradley Manning

One of the most recent and widely publicized examples of a malicious insider is Pfc. Bradley Manning. Pfc. Manning had a Top Secret/SCI security clearance during his deployment to Baghdad [26]. In 2010, he allegedly supplied WikiLeaks, a repository for whistleblowers to anonymously submit information, with 1.6 gigabytes of files, containing in excess of 250,000 diplomatic cables, logs from operations in Iraq and Afghanistan, and a video. A CD-RW was the tool Pfc. Manning used to exfiltrate data

from the Sensitive Compartmented Information Facility (SCIF) where he worked [27]. Pfc. Manning's actions are a breach of confidentiality, and the leaked documents are likely extremely damaging to the United States and its relationships with its allies. Pfc. Manning's ability to repeatedly burn files to a CD-RW demonstrates the lack of effective monitoring at the time he committed his malicious activities.

2.3.3 Aldrich Ames

Aldrich Ames was a spy for the Soviet Union for nine years in the 1980s and early 1990s [28]. During his service with the central intelligence agency (CIA), Mr. Ames supplied the Soviets with codes, techniques, financial spending, and the identities of several American double-agents [29]. Ames eventually provided the identities of all assets he knew to protect himself [28]. Of those whose names were supplied to the KGB, at least four are known to have been killed [29]. To obtain the information, Ames searched datasets from workstations that had floppy disk drives. Additionally, no protections were in place that restricted Ames to the least amount of information necessary to perform his job. He also had access to information not within his need-to-know [12]. At the time of his trial and conviction, he was referred to as the most damaging spy in the history of the United States [28-30].

2.3.4 Robert Hanssen

Robert Hanssen is a former FBI agent who sold information to the Soviet Union between 1985 and his arrest in 2001 [31]. During his career, he exfiltrated numerous top secret/sensitive compartmented information (TS/SCI) documents focusing on the USA's counter-intelligence program. Hanssen was extremely technical and used IT systems to assist in his malicious activities; he had developed several IT systems for the FBI during

his employment. To exfiltrate sensitive data, he would use floppy disks, removable storage, and a handheld computer [12]. Hansen used the databases he had access to perform keyword searches beyond his need to know to ensure he had not been detected. Searches included checking recent FBI entries, searching for his home address, or for a document drop location name. On several occasions, there were possibilities for him to be caught because of his actions on an IT system, but he remained undetected. In one such instance he was discovered with a password cracking program on his computer, but was able to avoid punitive action by claiming it was necessary to obtain the administrator password to install a printer. Ultimately, Hanssen was not caught as a result of his use of IT systems, but when his voice was identified [12] [31].

2.4 Virtualization

Within the Intel x86 architecture, there are four privilege modes or rings, which are numbered 0 to 3 and 0 is the most privileged. On a host OS, the operating system and kernel execute at ring 0 and applications at ring 3; rings 1 and 2 are not used. The separation of privileges allows the kernel and operating system to remain secure if an application should become compromised [35]. However, it is possible that ring 0 could become compromised and therefore the entire machine would be under an attacker's control. Virtual machines can assist with mitigating this threat.

In order to preserve the security the ring structure provides, Xen's paravirtualized environment runs the VMM at ring 0, the guest OS and kernel at ring 1 and guest programs at ring 3. The downside of paravirtulization is that it requires the guest to be modified, something that cannot be performed for Windows guests [35]. Full

virtualization, which is what will be used for this research, guests do not need modification. The hypervisor provides CPU emulation to the guest to allow privileged CPU functions. In both instances, the hypervisor are running at lower rings than the guest, therefore maintaining security between the guest and hypervisor.

A virtual machine is an isolated guest operating system instance running on a normal host operating system instance. A host operating system is able to run multiple virtual machine instances; the only limitation is the hardware resources available to the host operating system. Virtual machines have hardware abstraction performed through the hypervisor, or virtual machine manager (VMM). This allows VMs to function exactly the same as if they were a host OS. A VMM is designed as a small software layer to ensure isolation between virtual machines and the host system [32]. Additionally, the VMM allows users to specify the amount of virtual hardware, such as memory, available to a VM. Unlike host operating systems, VMs can be suspended and resumed without requiring a restart of the operating system. Suspending a VM pauses all currently executing processes and saves the guest OS's memory contents into the VM image file. This can be beneficial when a host needs to be restarted and the VM is critical to an organization's mission; the VM can be suspended, moved to a new host and restarted without negative impact to the organization's mission [33-34].

2.4.1 Benefits of Virtualization

One of the most obvious and valuable benefits is improved hardware utilization. A single host running a single set of services is not as efficient as running several virtualized operating systems with each running a single service. Additionally, user's workstations can be virtualized and connected via thin clients. Virtualizing a workstation

lets users have the ability to run programs that require a specific OS without giving them full control over a VMM on their desktop. Additionally, it allows organizations to combine several user workstations on a single host, resulting in improved hardware utilization [33].

As a testing environment, VMs provide a consistent system to programs, such as software or malware, to be developed. With snapshots, a VM can be quickly restored to a known good state in the event the guest becomes corrupted or unstable. This also allows for researchers to perform repeated experiments with similar outcomes because of the identical starting states.

2.5 Virtual Machine Introspection

Virtual machine introspection (VMI) is the process of externally examining a virtual machine "…for the purpose of analyzing the software running inside it."[37] This section describes motivation for VMI, the semantic gap problem between a hypervisor and the VM, and an overview of current VMI research and product capabilities.

2.5.1 VMI Motivation

With the recent increase in virtualization, organizations have looked for new techniques to monitor the security of their systems. VMI is emerging as a feasible and valuable method for securely monitoring a guest OS. Although bridging the semantic gap is challenging, VMI enables more secure guest OS monitoring. Software running within the guest OS is vulnerable to malicious insider modification or disabling while also remaining undetected to administrators. A user with full permissions to an OS instance can easily disable any security software with enough time, an abundant resource for

insiders. With VMI, users with full permission are unaware of the monitoring capabilities of the VMI tool and are also unable to compromise them. VMI allows the system administrators to continue to receive information about a VM despite the guest OS being compromised [37]. Additionally, a host-based intrusion detection system (HIDS) typically runs at user-level, meaning it can easily be compromised by malicious insiders or malware [37].

2.5.2 Semantic Gap

The semantic gap refers to the knowledge separation between a hypervisor and a guest VM. A hypervisor can easily observe raw memory values of a running guest, but turning that information into higher level data that is useful to an analyst or an automated tool is extremely difficult. Hypervisors cannot use API calls to the running guest without software running within the guest to serve as a middleman between the hypervisor and the guest OS. Without source code for the guest, an OS, such as Microsoft Windows, requires extensive reverse engineering of the guest's kernel and data structures to translate specific memory addresses into useful data. Additionally, patches to the guest OS or different version of the guest OS may change the data structure of an object, resulting in wasted reverse engineering effort [38].

Expanding further upon the inability to convert low-level system data into OS level information, obtaining context is also extremely difficult. Information such as employee's job position, work schedule, or project deadlines is valuable to an investigator in determining if a set of events are malicious or not [36].

One method for defeating the semantic gap involves paravirtualization. With a paravirtualized OS, the OS would supply information back to the VMM about programs

or data within the guest VM. This method requires modification of the guest OS in order to supply desired information. As a result, all closed source OSs, such as Microsoft Windows, cannot be paravirtualized. Additionally, since the guest VM is being trusted to supply correct information, it is susceptible to a malicious insider [84].

2.5.3 VMI Research

VMI provides a powerful platform to monitor processes and activity within the guest OS; a VMM can observe a guest while a compromised guest cannot disable the monitoring capabilities of the VMM. Much research has been performed on VMMs on detecting malware within a running guest. Unfortunately, most of these tools only address external system attackers and techniques they use to remain undetected from non-introspective anti-virus and forensic solutions. Malicious insiders typically do not employ malware for their actions and already have access to all of the sensitive information; therefore these tools are not well suited for detecting the insider threat.

2.5.3.1 Forensics

Forensics conducted through VMI has the added benefit of not requiring the target system to be shutdown; shutting down a production system can be very expensive for an organization. Additionally, shutting down the machine informs an attacker, either internal or external, that their actions have been detected. Research by [59] indicates that forensic analysis on a running and paused virtual machine provides approximately the same quality of results.

Research by [60] resulted in a VMI tool for Xen called Virtual Introspection tools developed for Xen (VIX). VIX was originally developed in order to overcome forensic challenges on running and powered off systems. On running systems, the use of forensic

27

analysis tools can result of a loss of critical data. Information such as the registry, network connections, logs, and temporary files are modified [60]. Additionally, running a tool can result in paging of memory out to disk. VIX provides similar functionality as Compiled Memory Analysis Tool – Virtual (CMAT-V), providing the capability to perform the following commands on a DomU system: ps, lsmod, netstat, lsof, who, and top. However, unlike CMAT-V, VIX pauses the target before acquiring the necessary data [60].

2.5.3.2 Static Forensic Analysis

In forensics, the traditional method for conducting investigations is static analysis. Static analysis has many existing policies and procedures to ensure captured data is legally valid. Typically the first step performed is to power down the compromised system [38-39]. Unfortunately, this immediately destroys valuable system information contained within the computer's memory. More importantly, if the system has an encrypted hard drive, the key is inaccessible once the machine is turned off. When shutting down a system, two methods are suggested: using the OS shutdown and pulling out the power cord. Both of these methods can contaminate the resulting disk image. Using the OS shutdown results in possible modification of logs, installing updates, or possibly malware that cleans up as the machine is shutdown. Pulling the power cord can result in an inconsistent or corrupt file system [38] [41].

2.5.3.3 Live Forensic Analysis

Live analysis remedies some of the problems arising from static analysis. Unlike static analysis, live analysis enables forensic investigators to obtain volatile information about the configuration of the system. The volatile information is often more useful when

performing forensic investigations. Volatile information on a system is kept in the memory and includes important information such as currently executing processes on the system, network connections, and contents of any files open on the system that have not been paged to the swap [42].

Although live analysis provides additional data to investigators, it also introduces some potential problems. First, malicious insiders could have running tools that hide critical information that would reveal their involvement in malicious activities. Since the tool and the forensic investigators tools are running simultaneously, the malicious tool can manipulate system responses provided to the investigator [38][43] Additionally, by examining the system, investigators may inadvertently cause the system to change state, causing subsequent data collections to not match the initial data capture.

Following Ken Thompson's axiom of "You can't trust code that you did not totally create yourself" [90], the next step for live analysis is leveraging VMI in order to preserve the volatile system data while ensuring the data collected is repeatable and not compromised. The introspecting process is running outside of the knowledge of the guest OS and therefore isolated from any tampering. Since the VMM controls all aspects of the virtual hardware presented to the guest OS, it can access all necessary volatile data an investigator requires [1].

2.5.4 XenAccess

XenAccess is a VMI and virtual disk monitoring tool used to monitor guest OSs running in Xen hypervisor. XenAccess' virtual disk monitoring is experimental and requires overcoming the semantic gap in to obtain useful data, meaning extensive reverse engineering would be required to monitor Windows disk activity. Xen runs guest VMs in

an unprivileged domain (DomU) and an administrative domain, which runs in privileged

domain 0 (dom0) [32]. XenAccess does not need to modify the VMM or the guest OS in

order to map memory of a DomU VM to a local address rage. XenAccess uses a structure

called `xa_instance` to maintain as much information about the DomU guest as

possible; this improves performance on subsequent calls. XenAccess then calls one of

three possible functions. An example of mapping a kernel symbol to a virtual address

provided by [32] is shown in Figure 2.1. `System.map` is a table of symbols and

addresses used by XenAccess to find a virtual address for the requested symbol.
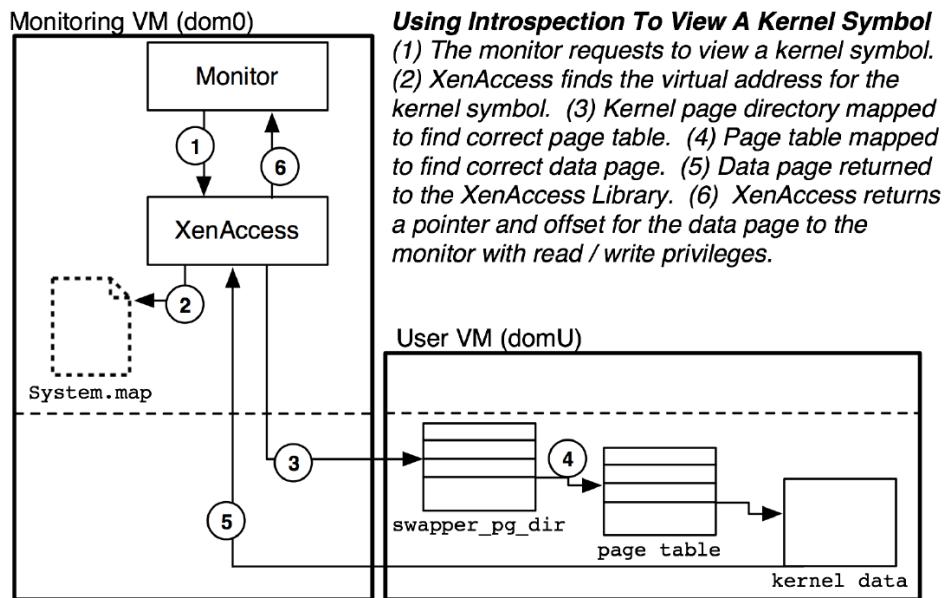
Figure 2.1: XenAccess Architecture [32].

2.5.5. Compiled Memory Analysis Tool – Virtual

Compiled Memory Analysis Tool – Virtual (CMAT-V) extends upon the

XenAccess VMI library and compiled memory analysis tool (CMAT). Unlike other

memory analysis tools, CMAT is able to be run against any Windows operating system

because of Program Database (PDB) files from the Microsoft Symbol Server in order to determine the location of specific application symbols [45].

CMAT analyzes memory dumps for system information such as network ports, active processes, drivers, registry keys, clipboard information and current users and it can save this information into feature files [45-46]. CMAT-V extends upon CMAT and is designed to perform live forensics upon a Windows DomU VM. Paravirtualization is not possible with CMAT-V because the tool is designed to run for Windows VMs and Windows is a proprietary operating system; necessary modifications to the guest operating system cannot be performed. The architecture developed by [47] is shown in Figure 2.2. CMAT-V utilizes Xen's hypervisor management API to interact with Dom0 and manage DomU virtual machines. CMAT-V was designed to run with CentOS version 5, but likely runs on any Linux distribution running a Xen kernel and with the necessary dependencies for XenAccess and CMAT-V.
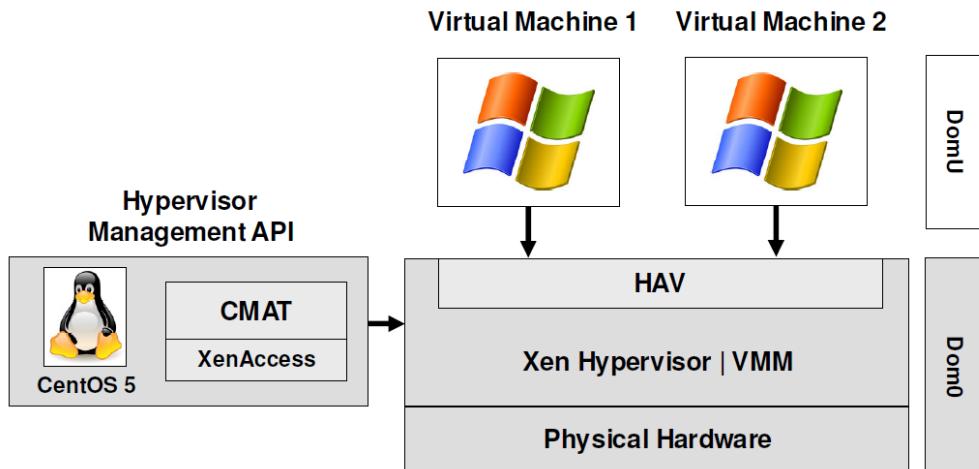
Figure 2.2: CMAT-V Architecture [47].

The most beneficial aspect of CMAT-V, for the purpose of this research, is its live introspection mode. This mode still generates the previously mentioned feature files and also produces full guest memory captures. The feature files are described in Table 2.X and by default, all of the files are created every thirty seconds, except for text file 5, which occurs at the same time as a full memory capture. Full memory captures are generated by default every thirty minutes and cause the creation of feature files to be suspended because CMAT-V is not multi-threaded. The impact on the guest while running virt-live mode was determined by [47] to be approximately 3% to 4.5% decrease in performance. On systems with multiple guests and instances of CMAT-V executing in virt-live mode, it can be expected that performance will continue to decrease because the Dom0's hard drive has more data generated by CMAT-V to write and each guest will have its own disk queries for normal system usage. This could be alleviated through the use of Redundant Array of Independent Disks (RAID) level 0 or Solid State Drives (SSD). Full memory captures enable forensic analysts to expand upon CMAT-V's feature files and either use existing tools or modify CMAT-V to obtain additional information from the guest's memory.

Table 2.3: Description of CMAT-V Feature Files.

Filename Suffix	Description
_1.txt	Lists currently running processes, including: process ID, process name and owner.
_2.txt	Lists current network connections, both TCP and UDP.
_3.txt	Lists currently loaded libraries, the path to the library on the system and the corresponding process ID.
_4.txt	Lists all current file handles, process ID, and the permission that process has to the file.
_5.txt	Lists process ID, and any registry entry corresponding to the process ID, to include the memory location for that registry entry.
_6.txt	Lists all currently loaded drivers and its base memory address.

2.6 Summary

This chapter summarized current insider threat research. A distinction was made between an insider and a malicious insider to create a definition to expand the research on. Insider threat characteristics, such as demographics, technical ability, and attack methods were presented. Next, several detection techniques were examined. The topics of virtualization and virtual machine introspection were introduced and the benefits they can provide. Lastly, the chapter described the tools which will be employed to develop an insider threat mitigation strategy.

III. Methodology

A malicious insider is capable of causing more damage to an organization because they are trusted individuals and function behind a majority of computer network defenses. This objective of this research is to determine if virtual machine introspection (VMI) can be leveraged to signal potential malicious insider threat behavior.

This chapter describes the formal methodology used to develop a proof-of-concept solution to malicious insider threats through VMI. The first section describes the problem, the goals of the research, assumptions which are made to limit the scope of the research, and the expected outcome. The second section describes a four step method for analysis of each malicious insider use case and corresponding generated scenario. The third section describes the test environment used to obtain malicious and non-malicious data. The last section describes problems and solutions encountered with Xen and Compiled Memory Analysis Tool – Virtual (CMAT-V).

3.1 Problem Definition

Existing malicious insider threat detection systems (ITDS) focus on the network level [83] [85], or execute at the same privilege level that the malicious insider is operating in [4] [77-78] [86]. Most host-based ITDSs are visible to the user therefore such systems can be subverted by an insider. Sensors placed on the network are unable to detect malicious actions that occur only at the user's workstation, such as copying files to removable media. Additionally, encryption can be used to bypass network level monitoring techniques. This research is motivated by the need to perform real-time analysis of a user's workstation while remaining undetected and executing at a higher

privilege level than the user. This provides security analysts with a more complete representation of the user's current actions.

3.1.1 Research Goals

The primary goal of this research is to determine whether insider threat detection can be performed on a Windows guest virtual machine (VM) through virtual machine introspection (VMI) using the CMAT-V. This research does not modify any files in the guest Windows operating system and uses existing capabilities of CMAT-V to perform full memory captures. The result allows guest VM introspection to remain transparent and inaccessible to the user. Additionally, the research also attempts to determine the successfulness of only generating alerts for malicious insider actions. Two additional data sets are used to validate the alerting methods. Although one or two observables within non-malicious data may cause an alert to be generated, all observables identified in a particular insider threat scenario should not be identified in non-malicious data.

3.1.2 Assumptions

The following assumptions are introduced to limit the scope of the research. These assumptions are independent of all use cases. If a use case requires specific assumptions, they are discussed in the description of each use case in Chapter 4.

- Users have full access to files on the DomU system, except for those specifically restricted by the Windows OS.

- Users are unaware of the existence of CMAT-V. As a result, malicious insiders will not attempt to obfuscate their activities from CMAT-V specifically, but may attempt to hide from DomU level monitoring. Previous work by [47] revealed a 3% to 4.5% decrease in performance within VMs while CMAT-V was executing;

users likely will not detect performance degradation and therefore will not detect introspection of their system.

- The host OS (Dom0), virtual machine manager (VMM), and CMAT-V cannot be circumvented, disabled, infected, or modified by the user.

- Threats modeled are intentionally malicious and their actions are not the result of an accidental breach of confidentiality, integrity, and availability (CIA).

- The malicious insider is acting alone and does not utilize social engineering tactics to aid their attack.

- The registry contained within DomU cannot be modified by the insider to hide their actions.

- Malicious insiders will not use physical attack to access other systems in the network.

- Workarounds for Xen USB passthrough and optical discs produce similar observables as native Windows functionality. Motivation for this assumption is provided in section 3.4 Experimental Limitations.

- Clipboard and print operations are performed on pre-determined files for both malicious and non-malicious users. Section 3.4.5 CMAT-V Limitations elaborates the rational for this assumption.

- All of the actions of a single malicious insider scenario are performed within the time span of one memory capture and each action is performed in the order specified.

- Documents deemed sensitive to the organization have been identified and appropriately flagged.

3.1.3 Expected Outcome

It is expected that through VMI of a Windows guest, specific registry keys and hexadecimal patterns within memory can be monitored for changes. When a change is observed, an alert is generated displaying the changed values, enabling an analyst to determine malicious intent. Additionally, an organization can mitigate potential disadvantages of alternative insider threat detection methods, such as cost or compatibility issues, by employing a tool relying on open-source code and executing outside of Windows workstations, preventing the tool from being compromised by the insider. Testing the alerting methods against data not containing a malicious insider, malicious insider attack patterns should not be identified. This outcome confirms the alert generation methods do not produce alerts for non-malicious user behavior.

3.2 Research Approach

To accomplish these research goals, a finite sequence of steps are developed and performed. Decomposing the research methodology into a precise sequence of steps allows the research to be repeated with the same results. The approach to the problem of insider threat alert generation through VMI involves four interrelated steps. The four steps are: development of malicious insider taxonomy, VMI observable analysis, malicious insider detection, and data validation. Prior to performing the four step process, six use cases are identified and decomposed into scenarios. Chapters 4 and 5 are organized by each use case and related scenario. The purpose of each scenario is discussed before addressing the four steps of the methodology.

3.2.1 Use Case Development

Use cases provide a high-level overview of actions a malicious insider could perform to achieve an unauthorized state of the system. The use cases are selected through examination of malicious insider techniques and security reports as discussed in Chapter 2. Each use case represents a malicious insider attempting to accomplish a malevolent objective differing from the organization's mission, such as theft of data or damage against an organization. Specific to each use case are several scenarios which are performed. These scenarios provide different techniques a malicious insider may employ and also allow the malicious steps to be decomposed into an attack taxonomy.

3.2.2 Malicious Insider Taxonomy

To accurately model and prevent malicious insider behavior, each scenario is decomposed into individual attack actions that can be observed from beginning to end. Decomposition of attacks enables VMI observables to be identified and an effective alerting strategy to be developed. As mentioned in Chapter 2, detailed information about real-world malicious insider incidents is not readily available; information is often summarized into preventative steps an organization should employ. The model described by [50] can be slightly modified and used to describe each malicious insider attack scenario. Each component of the taxonomy is described below.

Attacker. Malicious insiders are the attackers who perform actions against a company using information technology to accomplish an objective. A classification of these individuals is described in [50]. For the purpose of this research, the attacker is always a malicious insider. These individuals are already trusted users of the system and as such, attempting to classify them into the types of attackers

identified in [50] only differentiates the title of the attacker and not the tool, action, and target of their action.

Tool. A malicious insider begins their attack by using a tool to exploit some vulnerability within the system. A tool can range from a simple and legitimate command, such as copy and paste, to an automated program or virus. Multiple tools can be employed by a malicious insider during a single scenario.

Vulnerability. A vulnerability is a weakness or deficiency within an information system that can lead to unforeseen and unauthorized access [50]. A vulnerability is typically considered a bug in implementation of a software program that can lead to the development of an exploit. However, it can also be an architectural problem with the design of the system or a misconfiguration of the system.

Action. An action is a step taken by the malicious insider in order to obtain a desired effect. Actions incorporate the tool and vulnerability against the target in order to provide the desired result. Actions can include modification, deletion, disabling, moving, copying, pasting, installation, bypassing, or printing. Scenarios may include multiple actions by the insider.

Target. The target is the focus of the malicious insider's tool, vulnerability, and actions. A malicious insider's target is data on the system, or a running program on the system. Several example targets include sensitive corporate documents, other user's account credentials, and running programs on the system.

Unauthorized Result. An unauthorized result is defined to be the conclusion of the malicious insider's actions that is not permitted by the organization. These results

can include increased system privileges, denial of service, distribution of information, or modification of information.

Objective. The final item in the malicious insider taxonomy is the insider's objective. For the purpose of this research, knowledge of the objective is not relevant to successful detection, but possible objectives a malicious insider may have are enumerated. Objectives can include, but are not limited to, financial gain, damage, or espionage.

Tool	Vulnerability	Action	Target	Unauthorized Result	Objectives
Physical Attack	Design	Probe	Account	Increased Access	Challenge, Status, Thrill
Information Exchange	Implementation	Scan	Process	Disclosure of Information	Political Gain
User Command	Configuration	Flood	Data	Corruption of Information	Financial Gain
Script or Program		Authenticate	Component	Denial of Service	Damage
Autonomous Agent		Bypass	Computer	Theft of Resources	
Toolkit		Spoof	Network		
Distributed Tool		Read	Internetwork		
Data Tap		Copy			
		Steal			
		Modify			
		Delete			

Figure 3.1: Modified Computer and Network Incident Taxonomy [50].

Figure 3.1 describes the modified taxonomy developed by Howard, et al. [50]. Through modification of the computer and network incident taxonomy, the steps needed to successful complete an attack can be analyzed, starting from the tool employed by the

insider to the objective. This analysis will assist in determining indicators that can be observed through VMI.

3.2.3 VMI Observable Analysis

In the four-step research approach, the third step is identification of possible introspection observables. Each action in the scenario, as identified in the taxonomy, is individually analyzed and an identifier is recorded which facilitates successful observation. These identifiers consist of registry entries, hexadecimal patterns, or clipboard information. To identify observables, each action is performed within a Windows 7 virtual machine running procmon.exe to identify any possible changes in running processes or registry entries. If no observables are identified for an action using this method, a memory capture with an action is examined using a hex editor. Memory captures are examined for any unique hexadecimal patterns which would allow observation of an action. It is possible that a scenario may only have few or no observables through VMI and as a result, Windows event logs are employed to assist with identification of observables.

3.2.4 Malicious Insider Detection

The fourth step utilizes information obtained from the previous three steps to generate an alerting method for each scenario. Since the research focuses on VMI, observables identified in Section 3.2.3 are used for alerting methods, in combination with Windows event logs. Observables available within the guest that could improve detection accuracy are not used for insider alerting. Several scripts are developed to assist with the extraction of VMI observables from full memory captures. These scripts also compare changes between observables between different memory captures and generate an alert if

a change occurs, signaling potentially malicious behavior. Alerts highlight the changed items in different colors, based on the change from a previous memory capture. Green indicates a new entry, yellow indicates a change, and red indicates the entry no longer exists. After extracting the data from the full memory captures, the output is analyzed for each specific step in the scenario to determine if a single step can be declared malicious. For the purposes of this research, each malicious scenario has the actions performed in the order specified, thereby causing alerts to be generated in a specific order. Successful detection of a scenario requires alerts to be generated in the same order each action is performed, as listed previously in the assumptions section. Additionally, an entire scenario needs to occur between a single memory capture allowing each observable to generate an alert within the time span of one memory capture.

3.2.5 Data Validation

After developing detection techniques for each step in a scenario, the detection technique is compared against manually performed non-malicious scenarios and data collected from the Advanced Cyber Education (ACE) Hackfest. This allows the developed alerting mechanisms to be evaluated for accuracy in identifying only insider threats. Specific information regarding the non-malicious scenarios and ACE Hackfest can be found in sections 3.3.1.5 and 3.3.2, respectively. Analysis of generated alerts is expected to show several of the same observables identified in sections 3.2.3 and 3.2.4 are present, but do not indicate malicious activity.

3.3 Data Collection

For this research, two networks were used to capture data. The first data collection network is created specifically for this experiment to perform malicious insider scenarios. The second network used for data collection is one created during the ACE Hackfest. The experimental network also contained several non-malicious users which allows for collection of non-malicious data.

3.3.1 Malicious Insider Network Setup

For the insider threat experiment and data collection, the network is similar to the one used for the ACE Hackfest is constructed. CentOS 5.5 is used for the Dom0 OS and runs a Xen Linux kernel, 2.6.18-194.el5xen. The experiment uses two servers and five workstation VMs and are running on two Dom0 systems. The first host machine is a Dell Precision 690 desktop with 4GB of RAM, a 70 GB hard drive, and a dual-core Intel Xeon 5160 CPU running at 3.00 GHz. This machine runs two Windows 7 workstation VMs. The second physical machine is a Dell Precision M4500 laptop with 8GB of ram, a 450GB hard drive, and a quad-core Intel Core i7 M640 CPU running at 2.80 GHz. This machine runs three Windows 7 workstation VMs, and two Windows Server 2003 server VMs. These two machines are connected to a Linksys SD205 100 Megabits per second (Mbps) switch. This switch is connected to another network to obtain internet access for installing dependencies on the Dom0 machines and allow internet browsing on the workstation VMs. Figure 3.2 illustrates the logical configuration of the insider threat experiment network.
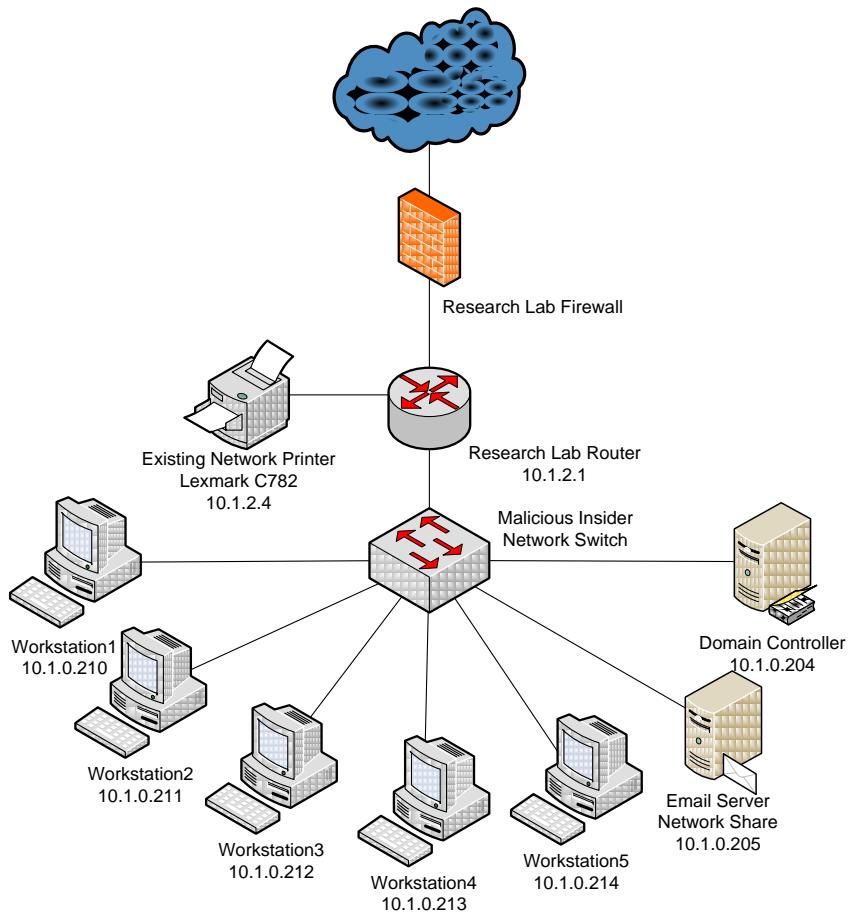
Figure 3.2: Logical Malicious Insider Network Design.

3.3.1.1 Servers

Server VMs consist of Windows Server 2003 with Service Pack 2 without additional security patches. The virtual hardware for the server VMs consists of 512 MB of ram, 10 GB of hard drive space, one virtual CPU, and are fully virtualized. One server runs Active Directory (AD) and a Domain Name System (DNS) server, and the second server runs the Microsoft Exchange mail server and a network file server. These services are found on many enterprise and government networks.

3.3.1.2 Workstations

Workstation VMs run Windows 7 without any security patches. Windows 7 is selected as the operating system for the workstations for two reasons; first, the Air Force's Standard Desktop Configuration will now use Windows 7 [68], demonstrating the relevance to the current Air Force mission. The second reason is Windows 7 provides consistency with the data captured during the ACE Hackfest. The virtual hardware for the workstations consists of 1024 MB of ram, 15 GB of hard drive space, one virtual CPU, and are fully virtualized. All Windows 7 workstations have Office 2007 Enterprise installed and are connected to a Windows domain running on one of the servers. Windows binaries often change when service packs or security patches are installed. Not installing any additional security patches ensures Windows data structures remain the same between both the ACE Hackfest and malicious insider data sets.

3.3.1.3 Windows Configuration

Changes are performed to the baseline Windows installation to enable features found in enterprise networks, and to allow additional attack methods for a malicious insider. All workstations and servers are configured to allow incoming Remote Desktop Connection (RDC). All workstations are connected to a networked printer. All workstations are configured to share their user's personal folder. For example, if a user is named mcrawford, the folder *C:\Users\mcrawford* is shared. Lastly, all users in the experiment are granted permission to all folders on the network file server.

3.3.1.4 CMAT-V Configuration

CMAT-V configuration is slightly modified for the malicious insider experiment. Instead of performing full memory captures every thirty minutes, delay between full

captures is reduced to ten minutes. Increasing the frequency between full captures enables collection of more data and less change can occur on the DomU systems with a shorter window. As a result, the malicious insider activity can be signed faster.

CMAT-V was also modified to accept an argument to create only full memory captures and only feature files. This enables two separate CMAT-V processes to be run against a single guest. Additionally, writing information to the feature files is no longer suspended during the full memory dump because there are two separate processes. CMAT-V is run in live introspection mode against all DomU machines and two instances of CMAT-V, one for feature files and the other for full memory captures, are run against each guest. The command to generate the full memory captures every ten minutes is listed below.

```
cmat -data <output_path> -memdump -feature <file_prefix>
-virt_live <VM_ID>
```

The six feature files are generated with a slightly modified set of command line arguments. The command used to generate feature files is listed below.

```
cmat -data <output_path> -feature <file_prefix> -virt_live
<VM_ID>
```

3.3.1.5 Non-Malicious Data

In addition to malicious insider scenarios, normal user behavior is performed within the malicious insider network. To generate non-malicious data, a script created by [67] is used. Some of the malicious insider scenarios performed by [67] are modified or omitted to maintain the research focus. In addition to this script, non-malicious scenarios are derived from the malicious insider scenarios. The purpose of performing actions

similar to malicious insider actions is to ensure only the insider threat actions are alerted on, and not normal user actions. The full script is presented in Appendices D and E.

3.3.2 Advanced Cyber Education

Advanced Cyber Education (ACE) is an eight week course at Wright-Patterson Air Force Base (WPAFB) held during the summer and is open to Air Force, Army, and Navy ROTC cadets. Participants are currently juniors and seniors and specializing in Computer Engineering, Computer Science, or Electrical Engineering. Subject matter during the course includes information warfare, computer network operations (CNO), digital forensics, reverse software engineering, and cryptography. The course is conducted at the Center for Cyberspace Research (CCR). The culmination of the course is a two day exercise focusing on CNO, where two teams attack and defend, while also performing typical user behavior, such as editing Word documents or sending email [55-56].

For this research, data from the ACE exercise is only used as an additional data set. Unfortunately, many actions simulated in the normal user data set are not present during the ACE Hackfest, such as USB activity, printing, or extensive file access. Actions that are performed are not well documented and assumptions used for this research, such as pre-identification of sensitive files, are not present for ACE data.

For the ACE Hackfest exercise, CentOS v5.5 was used for the Dom0 operating system running the Xen kernel. CMAT-V was configured to perform full memory captures of DomU virtual machines every thirty minutes and generating feature files every thirty seconds. DomU VMs consisted of two different types of machines. Servers for the exercise were running Windows Server 2003 with Service Pack 2, without any

additional security patches. These VMs were allocated 512 MB of ram. Workstations ran Windows 7 without any security patches and were allocated 1024 MB of ram. Security patches were not installed to improve the success of the attacks against the workstations and servers because only open-source tools were leveraged during the exercise. Documentation regarding additional software that may have been installed or used was not recorded. Additional machines on the network were attacker machines running BackTrack 5, a Linux distribution with many Computer Network Exploitation (CNE) tools. Attacker machines did not have CMAT-V running and are not included in the malicious insider test network.

3.4 Experimental Limitations

After initial research and reverse code engineering of the Windows printer, user-level and kernel level clipboards, it was discovered that CMAT-V cannot capture the information before the pointers are dereferenced. Additionally, several hindrances with Xen were encountered after the initial setup of the malicious insider network. This section describes the limitations encountered with Xen and CMAT-V, and the workarounds implemented to maintain a realistic corporate network. Information obtained from reverse code engineering the aforementioned Windows components can be found in Chapter 5.

3.4.1 Xen USB Support

VMMs allow configuration of virtual hardware to be presented to the guest as physical hardware. Examples of virtual hardware presented to the guest include network adapter, hard drive, processor, memory, sound cards, CD or DVD drives, and USB

devices. The version of Xen compatible with CMAT-V has very limited support for USB device passthrough to the guest and no support for passing through CD or DVD writing capabilities. The officially documented method for USB 1.1 passthrough to a fully-virtualized guest is using QEMU-dm [63]. QEMU is a machine emulator and virtualizer. In machine emulation, QEMU can run software designed for one type of machine on a different type. The virtualizer mode of QEMU allows execution of guest code directly on the Dom0 CPU [61]. The advantage QEMU-dm passthrough provides is it does not require additional drivers to be installed in the host or guest. Unfortunately, this method as documented by [62] [63] [64] only successfully passes a few devices to the guest. During setup of the experiment, only a USB printer was able to be successfully passed to the guest. Several external storage mediums including a flash drive, three external hard drives, an MP3 player, and an Android phone could not be passed to the guest. Although the printer appeared to be successfully passed to the guest, the device was not fully supported by Windows and when print jobs were sent to the printer, the Windows guest immediately displayed an error message.

3.4.2 USB Workaround

In order to generate realistic exfiltration scenarios, a solution to the limited USB interface support in Xen is needed. USBIP is a tool that was mentioned on the Xen Wiki page [63], so this was examined as a possible solution. Unfortunately, the server portion of the program required a Linux host and required somewhat extensive configuration. After testing several more programs, USB over Network [65] was determined to provide the necessary functionality with minimal setup. The disadvantage is the free version only allows one USB device to be connected by the client. USB over Network installs a driver

49

on the client and server machine to allow USB imitation. USB commands are encapsulated and transmitted over IP between the server and guest instances of USB over Network. During testing, external drives were presented to the guest as a local drive and a USB printer was able to receive and print documents.

3.4.3 Xen Optical Disc Support

Similar to USB support, Xen allows the passthrough of physical CD-ROMs to the guest, as well as passing an International Organization for Standardization (ISO) image to the guest. The guest is only given read permission to the aforementioned optical storage mediums and attempting to provide write support to the guest for these devices is not officially documented and has very little community discussion [66]. Several attempts were made to present the guest OS with a writeable optical disc, but the Windows guest always mounted a read-only optical disc.

3.4.4 Optical Disc Workaround

To allow a greater number of exfiltration scenarios, a solution to the lack of writeable optical discs is necessary. Researching this problem revealed many ISO generating tools, but few that support a virtual CD/DVD-RW. KernSafe's TotalMounter was chosen because it allows for discs to be created using the native Windows 7 CD/DVD-RW functionality [67]. Using the existing Windows 7 disc writing functionality is the most realistic scenario for exfiltrating data using a CD or DVD. Organizations are not likely to have additional tools installed for writing to CDs. For the purpose of the experiment, TotalMounter is used to mount an image file to CD/DVD-RW and then files are written to it using the Windows 7 burning functionality.

3.4.5 CMAT-V Limitations

The final limiting factor in this research is the current capability of CMAT-V. Extensive research was performed to obtain the user-level clipboard source and destination file path, as well as the kernel clipboard, move, and delete operations. However, CMAT-V currently does not perform captures frequently enough to obtain this information. This data is transient and can only be directly access while a specific function is executing on the processor. After execution finishes, it remains in memory, but can only be accessed through string searches. String searches for recovering clipboard contents are useful in forensic analysis only when it is known what is on the clipboard. In reality, it is impossible to perform string searches to determine what is on the clipboard. For this experiment, each clipboard operation is documented so string searches can be performed to simulate the ability to capture clipboard file operations.

In addition to the clipboard, extensive reverse engineering and kernel level debugging was performed to obtain information regarding print jobs. A similar limitation is faced regarding print jobs. The data remains in memory after the print job finishes, but it can only be accessed by kernel symbols while the function is being executed. Through analysis of several memory dumps, the following hex pattern was identified as being able to identify print jobs in memory captures.

```
4E005400 20004500 4D004600 20003100 2E003000 300038
```

Unfortunately, this hex pattern can appear approximately ten times even if only one actual print job is present, which requires additional analysis by a security analyst. It could not be determined why additional instances of this pattern occur in memory. Further examination revealed the following string to eliminate the false positives.

4E005400 20004500 4D004600 20003100 2E003000 30003800 00004D

However, this limits the captured print jobs to only Microsoft Office products; print jobs from programs not starting with the term Microsoft, such as Notepad, would be missed.

3.5 Methodology Summary

This chapter described the expected goals of the research and the methodology developed to perform and evaluate the research. The taxonomy to decompose each use case into a scenario was presented, along with the generation of observables, malicious insider detection and data validation. The third section discusses the experiment setup for the ACE Hackfest and the malicious insider networks. The last section discusses experiment limitations encountered in Xen and CMAT-V.

IV. Use Case Exposition

A malicious insider exists as an entity that is trusted by an organization and functions behind a majority of network defensive technologies. Their trusted position enables them to cause significant damage to an organization. Therefore, an improved mitigation technique which is transparent to potential insiders would greatly improve insider threat alerting capabilities.

This chapter discusses part of how the methodology created in Chapter 3 is implemented. The chapter is broken down by six use cases (UC) and presents the corresponding scenarios (S). Each scenario is broken down based on the malicious insider taxonomy, and a set of possible observables through virtual machine introspection (VMI) is generated. A summary of the VMI observables identified in this section is listed in Table 4.1. The structure for each section in this chapter consists of use case number, followed by the scenario number, and lastly the step number.

Table 4.1: VMI Observables Summary

Scenario	Description	VMI Observable
UC1.S1	Current Printers	HKLM\SYSTEM\ControlSet001\Control\Print\Environments\Windows NT x86\Drivers\Version-3
UC1.S1	Network Printers	HKLM\SYSTEM\ControlSet001\Control\Print\Monitors\Standard TCP/IP Port\Ports
UC1.S1	Current Printers	HKLM\SYSTEM\ControlSet001\Hardware Profiles\0001\System\CurrentControlSet\Control\Print\Printers
UC1.S1 UC6.S1 UC6.S2	Addresses typed in Windows Explorer	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\TypedPaths
UC1.S1 UC1.S3 UC4.S1	Recently mapped network drives	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU
UC1.S1 UC1.S2 UC4.S1 UC6.S1 UC6.S2	Recently accessed Word documents	HKCU\Software\Microsoft\Office\12.0\Word\File MRU

Scenario	Description	VMI Observable
UC1.S2 UC6.S1	Queries sent to Windows search	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery
UC1.S3 UC5.S1	Current user session info (W 7)	HKCU\Volatile Environment\1
UC5.S1	Current user session info (2003/XP)	HKCU\Volatile Environment
UC1.S3	Recent documents and shortcuts	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
UC2.S1	Microsoft Security Essentials Monitoring	HKLM \SOFTWARE\Microsoft\Microsoft Antimalware\Real-Time Protection\DisableRealtimeMonitoring
UC3.S1 UC4.S5	USB Device Information	HKLM\SYSTEM\ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
UC3.S1 UC4.S5	USB Device Information	HKLM\SYSTEM\ControlSet001\Control\DeviceClasses\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
UC3.S1	Mounted removable devices	HKLM\SYSTEM\CurrentControlSet\Enum\Storage\Volume\
UC3.S2 UC4.S2	Mounted network shares	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2
UC3.S2	CD Burning Information	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\CD Burning
UC3.S2	CD Burning Information	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\CD Burning
UC4.S2	Typed URLs in Internet Explorer	HKCU\Software\Microsoft\Internet Explorer\TypedURLs
UC4.S2	Mounted devices driver letter	HKLM\SYSTEM\MountedDevices
UC5.S1	RDP Information (Windows 7)	HKLM\SYSTEM\ControlSet001\Control\DeviceClasses\{28d78fad-5a12-11d1-ae5b-0000f803a8c2}\##?#Root#RDPBUS#0000# {28d78fad-5a12-11d1-ae5b-0000f803a8c2}\#TS001
UC5.S2	RDP Information (W2003/XP)	HKLM\SYSTEM\ControlSet001\Control\DeviceClasses\{28d78fad-5a12-11d1-ae5b-0000f803a8c2}\##?#Root#RDPDR#0000# {28d78fad-5a12-11d1-ae5b-0000f803a8c2}\#TS001
UC2.S3	InPrivate Browsing	49006E007400650072006E006500740020004500780070000 6C006F00720065007200200020002D0020005B0049006E00500 0720069007600610074006500 5D
UC1.S2	Print Jobs	4E005400200045004D004600200031002E00300030003800 0000
UC2.S3 UC4.S2	File Downloads	003A005A006F006E0065002E004900640065006E00740069 006600690065007200
UC2.S3 UC6.S2 UC6.S3	Browsing History	68007400740070003A002F002F00

Scenario	Description	VMI Observable
UC2.S3 UC6.S2 UC6.S3	Browsing History	0063006F006D005B0031005D002E00680074006D
UC6.S3	Email Contents	3C68746D6C20786D6C6E733A763D2275726E3A736368656 D61732D6D6963726F736F66742D636F6D3A766D6C222078 6D6C6E733A6F3D2275726E3A736368656D61732D6D69637 26F736F66742D636F6D3A6F66666963653A6F6666696365 22

4.1 UC1: Printing Activity

Printer use is a legitimate activity performed by a majority of computer users on a daily basis. However, a printer can be employed as a technique to exfiltrate sensitive or classified information by a malicious insider. In an environment without strict monitoring of employee's possessions when exiting the premises, a malicious insider could easily walk out with sensitive information. As discussed previously in Chapter 2 and by [68], disgruntled employees may use a printer as their method for stealing corporate data.

4.1.1 UC1.S1: Local Printer

This scenario examines a malicious insider who connects a new printer to their workstation. The first advantage presented to the insider by this technique is bypassing any network monitoring tools. Network printers are connected to workstations via Ethernet, which allows administrators to easily capture all or specific traffic items, such as print jobs. Another advantage the insider obtains though this method is bypassing monitoring methods on the printer itself. Tools such as [69] are often deployed within networks to monitor printer utilization and record job information. By directly connecting the printer to their workstation, the malicious insider is able to bypass both of these security features.

<u>4.1.1.1 UC1.S1.Step1: Taxonomy Development</u>

To more accurately identify observables in subsequent sections, the attack method is decomposed using the malicious insider taxonomy. This enables rapid identification of the actions performed by the insider and subsequently improved identification of VMI observables.

- *Tools:* For this scenario, the attacker uses several tools. The first tool is the Windows 7 OS. Servers are typically separated from printers through the use of virtual local area network (VLAN) and are physically secured in a locked room with locked server racks, so Windows Server OS is not analyzed for this scenario. The second tool the attacker uses is the printer itself. The printer is directly connected to the workstation by the malicious insider.

- *Vulnerability:* Several vulnerabilities can exist which would result in successful execution by the insider threat. One such vulnerability would be a lack of monitoring of USB ports on a user's workstation. This vulnerability allows malicious insiders to freely connect USB devices without an administrator's knowledge. Another possible vulnerability is relaxed policies regarding printer usage. IT staff may be willing to let users connect personal printers to their workstations without actual valid business reasons. For this scenario, the malicious insider will be exploiting both mentioned vulnerabilities.

- *Action:* The malicious insider performs the following steps in order for this scenario. First, the insider maps a network drive MS01\Organization\Project\Firewall to the Z drive locally. Next, the Xen USB

56

workaround program, USB-over-network, is opened. After opening the program, HP LaserJet 4350 is connected to the workstation. The Windows drivers automatically install and when installation is finished, the insider opens and prints Firewall Project Proposal from the previously mapped Z drive.

- *Target:* A malicious insider would likely target anything of potential financial value or anything that may be damaging to the organization if it were released to the public and/or an adversary. In this scenario, the malicious insider prints a Word document, *FirewallProjectProposal.docx*.

- *Unauthorized Result:* As a result of the malicious insider's actions, the user is now able to print documents at their own workstation without being detected by existing safeguards designed to prevent data exfiltration. Any documents printed can now be distributed without knowledge of the organization.

- *Objective:* The objective for this scenario is financial gain. The malicious insider has chosen to print *FirewallProjectProposal.docx* to reveal specifications about an upcoming project to an adversary.

4.1.1.2 UC1.S1.Step 2: VMI Observables

Identification of observables is critical for developing an alerting mechanism. The first observable, opening the USB-Over-Network program creates a new process on the system. This observable was omitted from alerting methods as the software is running only to emulate USB functionality for the Xen guest. This observable would not exist on separate hypervisors and may not exist on newer versions of Xen.

Several possible registry entries are identified for the action of connecting the printer to the workstation. The first is registry entry is `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Print\Environments\Windows NT x86\Drivers\Version-3`. This registry entry maintains a list of print drivers currently loaded on the system.

In addition to the *Version-3* registry entry, the entry of `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Print\Monitors\Standard TCP/IP Port\Ports` is monitored for changes. Since the printer was not connected via network, no additional entries were created, however if the malicious insider were to connect to a different network printer instead of using a local printer, it would be shown in this entry. In addition to the Ports registry entry, `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Hardware Profiles\0001\System\CurrentControlSet\Control\Print\Printers` is also monitored. In initial testing before the experiment, this appeared to have similar information as the Version-3 entry, but no changes are observed either before or immediately after performing this scenario.

The first registry entry evaluated is `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\TypedPaths`. This registry entry retains the twenty five most recent addresses typed into the Windows Explorer address bar. Monitoring this registry entry can assist organizations to ensure employees are staying within their work scope. During execution of this scenario, the value of this entry did not change; the malicious insider did not directly type the address into the Windows Explorer window.

In addition to the *TypedPaths* registry entry, `HKEY_CURRENT_USER\` `Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive` `MRU` registry entry maintains information about the most recently mapped network drives by the current user, in this case, the malicious insider.

The third action performed by the insider is opening the document *FirewallProjectProposal.docx*. By itself, this is not a malicious action and opening documents is an action performed by users of a computer multiple times per day. Additionally, the malicious insider's job description is software engineer and one of the projects he is working on is development of firewall software. The registry key `HKEY_CURRENT_USERS\Software\Microsoft\Office\12.0\Word\File MRU` maintains a list of the fifty most recently used (MRU) Microsoft Word documents; when a new file is opened, that file becomes *Item 1* in the list, and all other entries in the list have their item number increased by one.

The final action is observed through scanning full memory captures for a hexadecimal string. The pattern `4E005400 20004500 4D004600 20003100 2E003000` `30003800 00004D` successfully captures all Microsoft Office print jobs, but will not capture print jobs from non-office products such as Notepad. Therefore, a less precise pattern of `4E005400 20004500 4D004600 20003100 2E003000 30003800 0000` is used to scan memory captures for print jobs. This pattern generates several false positives, but will capture print jobs from any type of program.

Table 4.2: UC1.S1 VMI Observables.

	Insider Action Description	VMI Observable
1	Open USB-Over-Network	Running Process*
2	Connect Printer (HP LaserJet 4350dtn) to Workstation	Registry Entry
3	Map local drive Z to \\10.1.0.205\Organization\Projects\Firewall	Registry Entry
4	Open FirewallProjectProposal.docx	Registry Entry
5	Print Document to Local Printer	Memory Artifact

* Denotes this observable would not exist on a normal workstation and is therefore ignored.

4.1.2 UC1.S2: Work Scope Breach

This scenario examines a malicious insider who attempts to exfiltrate data not related to their position within an organization by searching for keywords. Motivation for this scenario is to determine if a work scope breach and printing the resulting document can be detected. This scenario uses the local printer discussed previously; for organizations that do not allow local printers, this scenario will have a similar taxonomy and observables for network based printers.

4.1.2.1 UC1.S2.Step 1: Taxonomy Development

The malicious insider threat taxonomy is again used to decompose this attack scenario to assist in identification of observables.

- − *Tools:* For this scenario, the attacker uses several tools. The first tool is the Windows 7 OS. As mentioned in the previous scenario, printing from a server is impractical for a malicious insider and would also be very suspicious if an individual carried a printer into the server room. Another tool employed the malicious insider in this scenario is the previously mentioned local printer.

- − *Vulnerability:* For the malicious insider to achieve the desired objective, they exploit a configuration vulnerability in the organization. This vulnerability is

lack of access control between different departments of the organization. This misconfiguration allows individuals to view all files on the organization's network drive.

Actions: The malicious insider performs the following actions in the listed order to accomplish the malevolent objective. First, the insider opens Windows Explorer. Next, the insider types the address *\\10.1.0.205* into the address bar. After connecting to the network drive, the insider uses Windows search functionality to search for "Classified". When the results of the query are displayed, the insider opens *AirForceBriefing.docx*. The attacker completes the scenario by printing the document to the USB printer.

− *Target:* Unlike the previous printing scenario, this scenario presents an instance where a malicious insider knows several of the projects occurring at the organization, but is not familiar with the details of the projects in other departments. The malicious insider will target classified information in other departments of the organization.

− *Unauthorized Result:* As a result of the insider's actions, the insider is able to access classified information which he is not authorized to access.

− *Objective:* The objective by the malicious insider in this scenario is either financial gain by selling the targeted information to a competitor or damaging the organization by publicly releasing the information.

4.1.2.2 UC1.S2.Step 2: VMI Observables

The insider's first action, opening Windows Explorer does not generate any observables. Explorer.exe process is always running on a Windows system and opening a new instance of Windows Explorer does not cause another explorer.exe process to be spawned.

Identification of navigating to a network drive is performed using the TypedPaths registry entry. As previously mentioned, this registry entry maintains a list of addresses typed into Windows Explorer taskbar.

The insider's next action is to search the network drive for "Classified". To facilitate detection of this action in the scenario, the registry key of `HKEY_CURRENT_USERS\Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery` is analyzed. Through analysis, it is determined that this registry entry stores the one hundred MRU Windows explorer search terms. When a new query is performed by the user, the each item in the list increases in number by one and the last item is removed from the MRU list.

The fourth action performed by the insider is to open *AirForceBriefing.docx*, the unauthorized document. After initially performing this scenario and a subsequent scenario, it was discovered that Compiled Memory Analysis Tool – Virtual (CMAT-V) had crashed sometime during this scenario. As a result, the unauthorized document access appears in both screenshots, but detection of this step can still be declared successful. Analysis of the output revealed the previously mentioned registry key of `HKEY_CURRENT_USERS\Software\Microsoft\Office\12.0\Word\File  MRU` would provide the necessary information for detection of this step.

The final action, printing the document, is detected via raw memory scanning of the previously identified hexadecimal pattern.

Table 4.3:UC1.S2 VMI Observables.

	Insider Action Description	VMI Observable
1	Open Windows Explorer window	None identified
2	Navigate to network drive by typing address in explorer window (\\10.1.0.205)	Registry Entry
3	Search network drive	Registry Entry
4	Open AirForceBriefing.docx	Registry Entry
5	Print Document to Local Printer	Hexadecimal pattern

4.1.3 UC1.S3: Suspicious Print Time

The final printing scenario examined involves a malicious insider printing an unusually quantity of documents outside of normal work hours. Recall from Chapter 2 that malicious insiders perform malicious actions outside of normal workplace hours to avoid detection by coworkers. Of the presented printing scenarios, this is likely the most damaging to an organization as the user has almost zero risk of being caught by a coworker if they are the only one in the office. Additionally, the insider has ample time to determine which documents are the most valuable and formulate a plan to avoid detection by any physical security at the building's entrance.

4.1.3.1 UC1.S3.Step 1: Taxonomy Development

- *Tools:* For this scenario, the attacker uses several tools. The first tool is the Windows 7 OS. Another tool employed the malicious insider in this scenario is the previously mentioned local printer.

- *Vulnerability:* For this scenario, the malicious insider is again exploiting the improper security configuration on the network drive. As described previously, the network drive is incorrectly configured to allow all users access to all files on the drive.

- *Actions:* To successfully execute the attack, the malicious insider performs the following steps in order. First, the insider accesses the workstation at a time outside of normal business hours for the organization. For this scenario, the access time is 00:02. Next, the insider maps a network drive to the organizations network file server. The folder *\\MS01\Organization* is mapped to the one of the insider's local drives. After mapping the network drive, the insider copies the five targeted files, Logger.cpp, PacketInspection.cpp, AutoUpdate.cpp, VM Configuration.xlsx, and Passwords.xlsx to the Desktop. The attack concludes with the insider printing the five documents to the local printer.

- *Target:* This scenario has the insider threat targeting the source code files for one of the projects at the organization and several other sensitive documents.

- *Unauthorized Result:* After performing the actions, the malicious insider is able to perform a disclosure of information to a third party.

- *Objective:* Similar to the previous exfiltration via printer scenario, the objective by the malicious insider is financial gain by selling the targeted

information to a competitor; either a competing company or adversarial nation-state.

<u>4.1.3.2 UC1.S3.Step 2: VMI Observables</u>

Prior to performing any malicious actions on the workstation, the user must first login to the system. Recall from Chapter 3 that all workstations are connected to a domain and as a result, all logon/logoff events are recorded both to the local machine and to the domain controller. This provides the organization with an advantage in that if the malicious insider is able to disable event logs on their own machine, some events are still recorded on the domain controller.

In addition to domain controller event logs, the registry entry `HKEY_CURRENT_USER\Volatile Environment\1` contains several subkeys which maintain information regarding the current user session. The `SESSIONNAME` subkey is set to `Console` when a user is currently connected to the system.

After logging on to the workstation, the malicious insider begins targeting several sensitive documents. To expedite this process, the user maps a drive to *\\ms01\Organization*, the hostname for the organization's network drive. Analysis of previous scenarios indicates `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU` is the registry key needed to detect this action. The malicious insider then copies five targeted files to the local desktop. This action is observed using brute force string search of the full memory captures.

The insider completes the attack by printing the five targeted documents to the local printer. Observation of this step is performed by searching full memory captures for

an aforementioned hexadecimal string. Additional observation is performed using the RecentDocs registry entry. Analysis revealed the registry entry `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs` maintains a list of the one hundred and fifty most recently accessed files by the user. Additionally, this folder also contains subkeys for each file extension the user has ever opened on the system. These subkeys also contain a MRU list and an MRUListEx value to indicate which items were accessed the most to least recently.

Table 4.4: UC1.S3 VMI Observables.

	Insider Action Description	VMI Observable
1	Login to workstation	Registry Entry
2	Map network drive	Registry Entry
3	Copy files	Clipboard
4	Open Document	Registry Entry
5	Print Documents	Hexadecimal pattern

4.2 UC2: Disable Defense Tools

This use case focuses on a malicious insider who has a technical background. As discussed in Chapter 2, some malicious insiders are technically proficient and may attempt to subvert known monitoring technologies. The motivation for this use case comes from the potential ability of a malicious insider to disable monitoring that is occurring on their workstation. The use of CMAT-V allows monitoring from a higher privilege level than the user and cannot be directly attacked or disabled unless the malicious insider is able to break out of the virtual machine, an undertaking that is extremely difficult.

4.2.1 UC1.S1: Disable Antivirus

For almost every organization, antivirus is the primary defensive tool employed on workstations against malware that is spread via email, browser exploits, or network exploitation. Newer HBSS may also employ user-level monitoring for insider threat actions. Depending on the specific product an organization uses to defend their workstations, this scenario may need to be modified to capture different observables.

4.2.1.1 Assumptions

Disabling certain antivirus or HBSS programs may require administrative privileges. This may help to reduce the number of individuals within an organization who are capable of disabling the protection on the workstation, but a determined individual could obtain this ability through several means, such as: collaboration with an external agent, social engineering of a coworker, or the user is already an administrator. Therefore, for the purpose of this research, it is assumed that the malicious insider has somehow obtained the required privilege to disable the antivirus.

4.2.1.2 UC2.S1.Step 1: Taxonomy Development

- *Tools:* For this scenario, the attacker a single tool, user commands. As stated in the assumptions, it is already assumed that the malicious insider has enough privilege to perform their actions.

- *Vulnerability:* The malicious insider is exploiting a design vulnerability within the antivirus product. Specifically, the antivirus has the ability to be disabled, which enables users to freely execute malicious programs on the system. This feature of an antivirus may be beneficial for a testing environment or personal

computer where suspicious programs are intentionally executed by the user, but in a corporate network under constant attack by external attackers, users should be prevented from disabling the antivirus.

− *Actions:* The insider performs the listed actions in order to accomplish the malicious objective against the target. First, the insider opens Microsoft Security Essentials. Next, the insider clicks the settings tab and selects "Real-time protection". The "Turn on real-time protection (recommended)" checkbox is unchecked and the insider clicks save changes.

− *Target:* A malicious insider is targeting the antivirus program in this scenario. Changing the properties of this process results in disabling this component of the workstation's defenses.

− *Unauthorized Result:* Upon disabling the antivirus, the malicious insider can perform a variety of tasks. Without any software to prevent malicious programs from executing, an attacker could cause a denial of service, corrupt or destroy valuable information, use their workstation to launch additional attacks against the network, or steal information that was protected by the HBSS.

− *Objective:* The objective of the attacker in this scenario is to mitigate any defensive technologies implemented by the organization to protect their workstation. After defeating the defensive tools, an attacker's objective may

be to steal documents for financial gain, hold certain data for ransom, or simply cause damage out of revenge.

4.2.1.3 UC2.S1.Step 2: VMI Observables

The first action performed by the malicious insider to disable the antivirus program is to open the program in order to access the settings. This step in the process did not provide any observables through VMI. Registry entries are typically not modified to indicate a program is open or closed, so instead the running processes on the system are examined. The process, `MsMpEng.exe`, provides the back-end functionality for the antivirus program. The graphical user interface (GUI) has a separate process, `msseces.exe`, but this process runs regardless of if the GUI is open or closed. Furthermore, changing to a different tab within the user interface did not create any possible observables.

This scenario only provides one possible observable for all actions, but arguably it is for the most importation action, disabling the antivirus. The registry entry `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft    Antimalware\Real-Time Protection\DisableRealtimeMonitoring` does not exist until the antivirus is disabled for the first time. Upon being disabled by the malicious insider, the entry is created and the value is set to one.

Table 4.5: UC2.S1 VMI Observables

	Insider Action Description	VMI Observable
1	Open Antivirus	None Identified
2	Navigate to Settings component of Antivirus	Non Identified
3	Disable Antivirus	Registry Entry
4	Open Document	Registry Entry

4.2.2 UC2.S2: Clear Windows Event Log

The Windows event logs are a valuable tool to administrators and security analysts to monitor activity on a system, such as installing software and account logons/logoffs.

4.2.2.1 Assumptions

Clearing Windows event logs requires administrative privileges. A technically proficient malicious insider could obtain administrator privileges through various techniques, such as: collaboration with an external agent, social engineering of a coworker, or the user is already an administrator. For this scenario, it is assumed that the malicious insider already has the necessary permission to clear the event logs. Another assumption is the event logs cannot be modified. Unlike Linux OS, Windows event logs are protected by the operating system and even administrators are unable to modify the event log; administrators can only clear all log entries. Linux treats the logs as a text file and any user with appropriate permission could remove individual entries from the log.

4.2.2.2 UC2.S2.Step 1: Taxonomy Development

- *Tools:* For this scenario, the attacker a single tool, user commands. As stated in the assumptions, it is already assumed that the malicious insider has enough privilege to perform their actions.

- *Vulnerability:* The malicious insider is exploiting a design vulnerability within the antivirus product. Specifically, the antivirus has the ability to be disabled, which enables users to freely execute malicious programs on the system. This feature of an antivirus may be beneficial for a testing environment or personal

computer where suspicious programs are intentionally executed by the user, but in a corporate network under constant attack by external attackers, users should be prevented from disabling the antivirus.

− *Actions:* To disable event logging on the insider's workstation, the following steps are executed in the order listed. The attack commences with the insider opening event viewer from the Start menu. The insider then expands the Windows Logs section and right clicks on Security. On the drop down list, the insider selects Clear Log. When prompted, the contents of the log are not saved. The insider repeats the steps to also clear the System event log.

− *Target:* A malicious insider is targeting the Windows event logs. By clearing the event logs, the insider is able to perform many tasks on the system and hide the evidence of their suspicious actions.

− *Unauthorized Result:* The immediate consequence of this scenario is the corruption of information; specifically removal of possibly incriminating log events on the user's workstation or server.

− *Objective*: The objective of the insider is to remove evidence of malicious activities performed on a workstation.

4.2.2.3 UC2.S2.Step 2: VMI Observables

The first step performed by the malicious insider is opening the Windows Event Viewer. The Windows Event Viewer contains several default event logs for a system including Application, Security and System. Events can be one of five possible types:

71

critical, error, warning, information, or audit success. Administrators can apply custom filters to identify specific event types or time periods to assist in troubleshooting a problem. Analysis of the running processes revealed `mmc.exe` to be the process that handles the Event Viewer GUI.

No VMI observables are able to be captured for the action of clearing the event log. The registry contains information regarding the event logs, but it does not have any information detailing when the log is cleared. Additionally, no new processes are created during this event.

Table 4.6: UC2.S2 VMI Observables.

	Insider Action Description	VMI Observable
1	Open Windows Event Viewer	None Identified
2	Clear Event Log	Windows Event Log

4.2.3 UC2.S3: Private Browsing

Private browsing is a feature in most modern browsers, including Internet Explorer, Google Chrome and Mozilla Firefox. The purpose of this functionality is to prevent history and multimedia items from being stored on the local computer. Although this does not prevent network level traffic inspection, a malicious insider could use this in combination with either HTTP Secure (HTTPS) or Secure Shell (SSH) to bypass network level defenses and possibly hinder post-incident forensics. For the purposes of this scenario, only Internet Explorer will be evaluated as most organizations do not allow users to install additional software on their workstation.

<u>4.2.3.1 UC2.S3.Step 1: Taxonomy Development</u>

- *Tools:* The malicious insider leverages two tools during this scenario. First, the insider uses Internet Explorer and the private browsing mode. Lastly, the insider uses built-in user commands.

- *Vulnerability:* The insider is exploiting a configuration vulnerability within the network. Windows Group Policy is capable of preventing users from accessing this feature within Internet Explorer, and private browsing should be blocked.

- *Actions:* To subvert potential workstation forensics, the insider performs the following steps in order. The insider begins by opening Internet Explorer (IE). Next, IE is switched to InPrivate browsing mode using Ctrl + Shift + P. The insider then navigates to http://www.darkcomet-rat.com. Due to some sort of bug, the download does not work and the insider goes to google.com and searches for "poison ivy hack". The insider clicks the first link leading to http://www.poisonivy-rat.com. The latest version, Poison Ivy 2.3.2, is downloaded by the insider and saved to C:\Users\tgreen\Downloads\PI2.3.2.rar.

- *Target:* The target of the malicious actions is the user's own computer. By enabling private browsing, the insider is attempting to minimize the forensic artifacts resulting from web browsing.

- *Unauthorized Result:* The consequence of the actions by the insider is corruption of information. In particular, some of the forensic artifacts typically remaining from web browsing are not written to disk while private browsing is activated. This limits the ability of a forensic investigator to recover the user's actions.

- *Objective:* The malicious insider's objective is to minimize forensic artifacts left from downloading a piece of malware.

4.2.3.1 UC2.S3.Step 2: VMI Observables

The first action performed by the insider is opening Internet Explorer. This action can easily be observed through monitoring the running processes on the system. However, this is a very normal action and performed by a majority of users on a daily bases. Attempting to differentiate malicious from non-malicious intention through this action would be extremely difficulty.

The second action the insider performs is switching Internet Explorer to InPrivate mode. No registry entries or processes could be identified to be modified as a result of the switch to InPrivate mode. Instead, a brute force search through memory is performed for the pattern listed below. This pattern is the hex representation of "I.n.t.e.r.n.e.t. .E.x.p.l.o.r.e.r. .–. .[.I.n.P.r.i.v.a.t.e.]", with the periods representing null characters.

```
49006E007400650072006E00650074002000450078007000 6C006F00720065007
20020002D0020005B0049006E00500072006900760061007400 65005D
```

The next action performed by the malicious insider is to navigate to several remote administration tool (RAT) download sites. Since the insider is using private

browsing, no registry entries are recorded for the user directly navigating to these pages. Instead several patterns are identified which allow limited detection of browsing history, while generating some false positives. The identified patterns are listed below

```
68007400740070003A002F002F00
```

```
0063006F006D005B0031005D002E00680074006D
```

The last action performed by the malicious insider is downloading the RAT. Again, no registry or process observables are identified, so a pattern based brute force search is required. The pattern is listed below.

```
003A005A006F006E0065002E004900640065006E00740069006600690065007200000
```

Table 4.7: UC2.S3 VMI Observables.

	Insider Action Description	VMI Observable
1	Open Internet Explorer	Running Process
2	Switch to InPrivate Browsing	Hexadecimal Pattern
3	Visit RAT Websites	Hexadecimal Pattern
4	Download RAT	Hexadecimal Pattern

4.3 UC3: Removable Media

As discussed in Chapter 2, removable media is another frequently used method for stealing sensitive data from an organization. The Department of Defense (DoD) currently bans removable flash media, such as USB thumb drives, from all Unclassified but Sensitive Internet Protocol Router Network (NIPRNET) computers [70]. However, as demonstrated by a use case in Chapter 2, malicious insiders will find alternate methods to exfiltrate data while still adhering to DoD policy. This use case addresses malicious insiders who use DoD approved removable media to steal sensitive information.

4.3.1 UC3.S1: External Hard Drive

The most obvious alternative to a USB flash media is a USB hard drive. Both provide similar functionality and are fairly compact in size; a USB hard drive could easily be hidden inside a stack of papers, briefcase, or a shoe to bypass physical security inspections. This scenario examines an insider who uses a USB hard drive to steal a document contained within the insider's work scope.

4.3.1.1 UC3.S1.Step 1: Taxonomy Development

− *Tools:* The malicious insider employs several tools in this scenario. First, normal user commands are performed to copy the file to an external hard drive. The final tool is the external hard drive itself. Due to limitations with Xen, a software workaround is used to simulate direct connection of the USB drive to the workstation. The program used to facilitate this is discussed in Chapter 3, but will not be listed as a tool in this section so this scenario remains representative of a real-world attack.

− *Vulnerability:* The insider is exploiting a policy vulnerability in this scenario. Although removable hard drives are effective for transferring data between computers, they introduce a great advantage for a malicious insider. The policy should prohibit all removable hard drives, or if they must be used, they need to be returned on the same day the drive is loaned out.

− *Actions:* To successfully exfiltrate data via a USB hard drive, the insider performs the following actions in the sequence listed. The insider first opens USB-over-network and connects a Western Digital My Passport external hard

drive to the workstation. The drivers for this drive are automatically installed by Windows 7. After the drivers are finished installing, the insider copies FirewallSource.zip from C:\Users\tgreen\Desktop to the external hard drive. The scenario concludes with the insider disconnecting the external hard drive from the workstation using USB-over-network.

− *Target:* A malicious insider is targeting the any sensitive information that will provide financial benefit. In this scenario, the malicious insider is targeting the source code to a firewall project. The insider is the developer, so accessing the source code is not suspicious.

− *Unauthorized Result:* The effect of the malicious actions is disclosure of information. The organization's confidential documents can be released by the insider to individuals who are not authorized by the organization to possess this information.

− *Objective:* As illustrated in Chapter 2, the malicious insider's objective or motivation may not be purely financial; the insider may perceive themselves to be a whistle blower on corruption and seek to disseminate information that ultimately damages the organization.

4.3.1.2 UC3.S1.Step 2: VMI Observables

Ignoring the Xen USB workaround, the first action performed by the malicious insider is connecting the external hard drive to the workstation. During initial analysis,

several observables are discovered that may indicate a change in currently connected removable media devices.

Several registry keys are examined for detecting the malicious insider's first action of connecting the external hard drive to the workstation. The first key is `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{53f56307 -b6bf-11d0-94f2-00a0c91efb8b}`. This registry key records all devices connected to the system and also enables analysis to determine when the last device was connected to the system, based on the last updated timestamp.

In addition to the aforementioned registry key, the registry key `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{53f5630d -b6bf-11d0-94f2-00a0c91efb8b}` maintains a similar list of devices connected to the system.

A small discrepancy was discovered between the Xen workaround for external drives and physically connecting them to a workstation; physically connected drives have the prefix `##?#STORAGE#VOLUME#_??_USBSTOR#DISK&`, whereas the mounted external drive only contained the prefix `##?#STORAGE#VOLUME#`. Recall from Chapter 3 that the Xen workaround causes the external drive to be mounted as a local volume to the guest instead of removable media.

The final registry entry changed as a result of the first action is `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Storage\Volume\`. This registry entry contains subkeys listing what appears to be a unique identifier for each storage device.

The second action performed by the insider is to copy the file to the clipboard and paste it to the external hard drive. Due to the aforementioned Xen limitations, this detection is limited to a brute force search through the full memory capture. The ability to detect clipboard file operations is discussed in Chapter 5.

The final action performed is disconnecting the external USB hard drive from the workstation. Detecting this action allows a full timeline of events to be created and provide further details for a security analyst to determine if the actions are malicious or benign. No observables were detected for this action in the scenario.

Table 4.8: UC3.S1 VMI Observables.

	Insider Action Description	VMI Observable
1	Connect USB hard drive to workstation	Registry Entries
2	Copy files to clipboard	Clipboard
3	Disconnect USB hard drive	None Identified

4.3.2 UC3.S2: Optical Disc

Another alternative to USB flash media is an optical disc, such as a Compact Disc-Recordable (CD-R), Compact Disc-Rewritable (CD-RW), or DVD-Recordable (DVD-R). Although these devices are not as easy to hide as an external hard drive and do not store as much information, they are still a useful storage medium for a malicious insider to exfiltrate data.

The most obvious alternative to a USB flash media is a USB hard drive. Both provide similar functionality and are fairly compact in size; a USB hard drive could easily be hidden inside a stack of papers, briefcase, or a shoe to bypass physical security inspections. This scenario examines an insider who uses a USB hard drive to steal a document contained within the insider's work scope.

− *Tools:* Similarly to the external storage scenario, two tools are employed in this scenario, excluding the Xen workaround. First, normal user commands are issued. The final tool is the external hard drive itself. Due to limitations with Xen, a software workaround is used to simulate direct connection of the USB drive to the workstation. The program used to facilitate this is discussed in Chapter 3, but will not be listed as a tool in this section so this scenario remains representative of a real-world attack.

− *Vulnerability:* A policy vulnerability is exploited by the malicious insider in this scenario. An organization's files can be extremely sensitive and steps need to be taken to ensure users cannot transfer data to any form of optical disc or removable media. As previously discussed, optical discs should only be allowed in rare circumstances.

− *Actions:* The insider's attack commences by mapping the remote drive \\workstation3\Users\lscarlet to X. Subsequently, the insider copies the files *NewHire.docx, Payroll.xlsx and SocialSecurityNumber.xlsx* from the X drive to his or her C drive. The insider then opens KernSafe TotalMounter and creates a virtual CD-RW. Windows is then presented with a burnable CD and a new CDRom drive. The insider opens Windows Explorer and opens the newly blank CDRom. The aforementioned files are copied from the C drive to the CDRom folder. Next, the insider burns the files to the CD using the Windows burn functionality. The attack ends when the burn is complete.

- *Target:* In this scenario, the malicious insider is targeting human resources information contained on another user's workstation. Specifically, the files *NewHire, Payroll and SocialSecurityNumber* are targeted by the insider.

- *Unauthorized Result:* The direct result of the insider's actions is disclosure of information. Additionally, the insider has unauthorized access to the user's documents on workstation 3. These documents are confidential and the organization does not want them to be disclosed publicly or to a competitor.

- *Objective:* The objective in this scenario is to steal the sensitive information from the organization.

4.3.2.2 UC3.S2.Step 2: VMI Observables

To identify the first action, the aforementioned Map Network Drive MRU registry entry is monitored. As previously mentioned, this MRU list maintains a list of the most recently mounted network drives or shares. This registry entry enables alert generation for mapping a network drive. Additionally, the registry key `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2` maintains a list of all mounted volumes, including CDROMs and drives connected via *net use* user command and is used to verify the information obtained from the network drive MRU.

The second action, copying several files from the mounted network drive, is observed using the Windows clipboard. Monitoring the clipboard for file copies allows security analysts to have detailed knowledge regarding file transfer operations occurring on a user's workstation.

Several markers are identified within the Windows registry to enable monitoring of CD burning. First, `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\CD Burning` contains information regarding the drive capable of writing to a CDRom. Additional analysis of the burning process revealed several changes which occur to this registry entry and allow for observation of this action. The second registry entry used to confirm the burning process is occurring is `HKEY_CURRENT_USERS\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\CD Burning`. Analysis of this registry entry reveals it contains the same information as the aforementioned …\Explorer\CD Burning registry entry. No observables are identified to verify exactly which files are burned, other than examining the clipboard.

Table 4.9: UC3.S2 VMI Observables.

	Insider Action Description	VMI Observable
1	Map Network Drive	Registry Entry
2	Copy Files	Clipboard
3	Burn CDRom	Registry Entries

4.4 UC4: Employee Behavior

Sudden changes in employee behavior are frequently a precursor to malicious insider attacks against an organization. Recall from Chapter 2 that coworkers often observe visible warning signs from the insider before malicious actions are performed against the organization. These scenarios attempt to address several suspicious employee behaviors which an organization could monitor to assist in mitigating insider attacks.

4.4.1 UC4.S1: Unauthorized File Access

The first employee behavior scenario is unauthorized file access by the malicious insider. This scenario is representative of an employee who is able to obtain access to a file that is not within their job description. For the purpose of this scenario, it is ignored how the access was obtained. Possible methods for access being obtained are through privilege escalation or incorrectly configured permissions.

4.4.1.1 UC4.S1.Step 1: Taxonomy Development

− *Tools:* The malicious insider only performs normal user commands to achieve the malicious objective.

− *Vulnerability:* The exploited vulnerability is a result of a configuration error within the organization. Specifically, every folder on the network drive containing many sensitive documents can be accessed by any user with a domain account.

− *Actions:* The attack starts with the insider maps the Y drive to the folder \\10.1.0.205\Organization\Mustard\Performance Reviews. The insider then navigates to the mapped network drive using Windows Explorer. The insider the copies all listed performance reviews, Crawford-2011.docx, Green-2011.docx, Green-2012.docx, Peacock-2011.docx, Scarlet-2011.docx, and White-2011.docx. The insider pastes all of the documents to his or her local workstation in the directory C:\Users\tgreen\Desktop. The attack concludes with the insider opening White-2011.docx.

- *Target:* The malicious insider is targeting the performance evaluation of a coworker who receives a higher salary than the malicious insider. The targeted information is contained on the organization's network drive in the CEO's personal folder.

- *Unauthorized Result:* The effect of the insider's actions is unauthorized access to the CEO's performance evaluations. The CEO's confidential documents are disclosed to the insider without approval from the documents' owner.

- *Objective:* Obtaining the performance evaluation allows the insider to discover the pay information for other employees and compare that information against industry averages.

4.4.1.2 UC4.S1.Step 2: VMI Observables

The first action performed by the insider is to map a drive to the CEO's folder on the organization's network drive. The aforementioned registry entry of `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU` maintains a list of currently mapped network drives and the corresponding letter on the local workstation. This registry entry enables observation of the first action.

The Windows clipboard is observed for detection of the second action; copying and pasting the performance evaluations from the network drive to the insider's desktop.

The final action performed by the insider is to open the Microsoft Word document White-2011.docx, which is copied to the desktop. The previously identified Microsoft

Word File MRU list is the observable monitored to alert for the action within the malicious insider scenario.

Table 4.10: UC4.S1 VMI Observables.

	Insider Action Description	VMI Observable
1	Map Network Drive	Registry Entry
2	Copy Performance Evaluations to Desktop	Clipboard
3	Open White-2011.docx	Registry Entry

4.4.2 UC4.S2: Unauthorized Software

This scenario models an insider who installs additional software on their computer to assist with data exfiltration. An insider who is able to install software can use the installed to subvert existing defensive technologies employed by the organization on the network and/or workstation.

4.4.2.1 UC4.S2.Step 1: Taxonomy Development

- *Tools:* The malicious insider uses Internet Explorer to obtain the software. Additionally, the insider uses the unauthorized software, TrueCrypt, to exfiltrate the data. Lastly, the insider performs normal user commands to access the desired data.

- *Vulnerability:* The exploited vulnerability is a result of a configuration error of the domain. Users are allowed to install software on their workstation without administrator approval. However, even if users are prevented from installing software, some software does not require administrator rights and can be run without installation, so the vulnerability remains. To completely

eliminate this vulnerability, an organization would need to employ process whitelisting.

- *Actions:* The actions performed in this scenario are completed in the order listed. The first action performed by the insider is to open Internet Explorer. Next, the insider types http://www.truecrypt.org into the address bar. After accessing the website, the insider downloads and installs a default installation of TrueCrypt. An encrypted volume is then created and mounted as the E drive. The malicious insider copies Firewall Project Proposal.docx to the TrueCrypt volume. The scenario ends when the insider dismounts the TrueCrypt volume.

- *Target:* The malicious insider is targeting the FirewallProjectProposal.docx document. This document is within the insider's work scope, so accessing it is not suspicious.

- *Unauthorized Result:* By performing this attack, the malicious insider causes the document to be disclosed to individuals who are not authorized by the organization to view its contents.

- *Objective:* The likely objective for the malicious insider in this scenario is financial gain or damage against the organization. Providing the document to a competing organization or nation-state would provide financial recompense for the insider. Releasing the information to the media would likely cause damage against the organization's reputation.

<u>4.4.2.2 UC4.S2.Step 2: VMI Observables</u>

Step one in the attack is opening Internet Explorer. This action can easily be observed through monitoring of the process list.

The next action performed by the insider is navigating to TrueCrypt.org. The script developed to detect browsing history is likely the method to observe this action. Additionally, the registry entry `HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\TypedURLs` maintains a twenty-five MRU list of URLs typed directly into the address bar. If a user accesses a URL via a link on a page, such as a search engine's results, the URL will not be recorded in this MRU list.

After navigating to the website, the malicious insider downloads TrueCrypt. This action can be monitored using the previously created script to scan memory captures for file downloads.

Installation of the program can be monitored through analysis of the currently running processes on the system. An organization employing either blacklisting or whitelisting could rapidly detect this action, unless the process is renamed to a common process name, such as iexplore.exe, the process name for Internet Explorer. Registry entries are also often created during the software installation process, so an organization could compare a clean registry snapshot with a current snapshot to detect changes; however, this technique is not used for detection in this scenario.

The fifth action performed by the malicious insider is creation and mounting of an encrypted TrueCrypt volume. Following a typical TrueCrypt encrypted file container setup results in a mounted volume showing up as a local drive and not a removable device. Therefore, previously identified VMI observables for external hard drives are not

applicable for this action. Analysis of TrueCrypt's behavior reveals observables for this action are similar to connecting an external storage medium. To support detection of this action, the previously mentioned registry entry …\Mountpoints2 and `HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices` are monitored. The registry entry …\MountedDevices is a list of devices that have been mounted on a system and is stored in binary form. This list includes local system drives, such as the C drive.

The sixth action performed by the malicious insider is copying FirewallProjectProposal.docx from a network drive to the TrueCrypt volume. Detection of this step is performed by monitoring the clipboard for file copy and paste operations.

The final step performed by the malicious insider is to dismount and re-encrypt the TrueCrypt volume. The previously identified …\MountedDevices registry entry is determined to provide an observable for this action.

Table 4.11: UC4.S2 VMI Observables.

	Insider Action Description	VMI Observable
1	Open Internet Explorer	Running Process
2	Navigate to truecrypt.org	Registry Entry
3	Download TrueCrypt	Hexadecimal Pattern
4	Install TrueCrypt using default settings	Running Process
5	Create and Mount TrueCrypt Volume	Registry Entry
6	Copy FirewallProjectProposal.docx from network drive	Clipboard
7	Dismount TrueCrypt Volume	Registry Entry

4.4.3 UC4.S3: Suspicious User Command – FTP

This scenario is representative of an insider who uses existing Windows functionality to exfiltrate data to a remote machine.

− *Tools:* The malicious insider uses only existing Windows functionality. Unlike the previous scenario, this scenario could prove more difficult to detect as there is not an obvious tool download and execution by the user.

− *Vulnerability:* For the insider to achieve the desired objective, a configuration vulnerability is exploited. The organization does not prevent users from using the existing file transfer protocol (FTP) functionality found within Windows. This configuration vulnerability could be difficult to detect by an organization since the tool is completely contained within the OS.

− *Actions:* To successfully exfiltrate the data, the malicious insider performs the following steps in the order listed. First, the insider opens a command prompt. The insider then navigates to the desktop using the command *cd Desktop*. Next, the *mkdir files* command is issued to create a folder on the desktop named files. The insider then changes directories using *cd files*. Once in the directory, the files are copied to this folder in preparation for exfiltration using *copy z:\FirewallSource.zip c:\Users\tgreen\Desktop\files*. The insider is not extensively familiar with the Windows ftp command and first issues *ftp --help* to learn more about the command. The insider then issues *ftp martincrawford.net* to connect to a remote server via ftp. When prompted, the insider supplies a known password. The command *mput* is used to put the zip file onto the remote machine and the insider finishes the scenario by issuing the *quit* command.

89

– *Target:* The insider is familiar with the organization's firewall project and knows the value is extremely high. Therefore, exfiltrating and subsequently selling the data would be extremely profitable.

– *Unauthorized Result:* Performing this attack results in disclosure of the source code by the insider to a third party not authorized to have access to the data.

– *Objective:* The insider's objective is to sell the source code to a competitor for a large financial reward.

4.4.3.2 UC4.S3.Step 2: VMI Observables

Unlike many other scenarios performed in this experiment, the insider exclusively relies upon the command line. Initial analysis resulted in several strings which generate an alert for command line ftp activity; no registry values are determined to change as a result of the previously listed actions. However, the Volatility framework developed a solution which is able to capture command line history. This tool is used in combination with the full memory captures to observe a user's command line behavior. Windows 7 command line history is much more difficult to obtain because as soon as the cmd.exe process is terminated, all history from the current cmd process is lost. Previous versions of Windows use csrss.exe to maintain command line history and the history persists even if cmd.exe is terminated [71].

Table 4.12: UC4.S3 VMI Observables.

	Insider Action Description	VMI Observable
1	Command Line Commands	Command Line History

4.4.4 UC4.S4: Suspicious User Command – File Deletion

This scenario is representative of a malicious insider who is motivated by revenge and the desire to cause damage to the organization. Unlike other scenarios, the insider is not attempting to steal property from the organization. Instead, the only goal is to destroy data within the organization.

4.4.4.1 UC4.S4.Step 1: Taxonomy Development

− *Tools:* Similar to the previous scenario, the malicious insider uses only existing Windows functionality.

− *Vulnerability:* The insider exploits a misconfigured network share on workstation 3 to perform the attack.

− *Actions:* The malicious insider performs the following actions in order to attack the organization. First, the insider uses the command *net use* to list currently mapped network drives. Next, the insider issues the command *x:* to change to the x drive, a network drive previously mapped to workstation3. The insider lists the contents of the current directory with the *dir* command. Three files exist in the directory, funnypic.jpg, german_shepherd_dog_664_12.jpg, and Social SecurityNumbers.docx. The insider completes the scenario by issuing the *del* command to delete each file individually.

- *Target:* The insider has previously mapped a network drive to workstation3 and seeks any files on the target computer to attack.

- *Unauthorized Result:* The effect of the insider's attack is a denial of service against the organization by deleting a critical file.

- *Objective:* The objective of the insider is motivated by revenge against both the organization and the user of workstation3. The insider seeks to destroy any targets of opportunity found within workstation3.

4.4.4.2 UC4.S4.Step 2: VMI Observables

As mentioned previously, the Volatility project has a plug-in for their framework which reliably extracts command line history than a brute force string search. A string search is only effective when it is known what commands are issued by the user. No registry entries, save for the previously mapped network drive, are identified for assistance in detection of the insider's actions.

Table 4.13: UC4.S4 VMI Observables.

	Insider Action Description	VMI Observable
1	Command Line Commands	Command Line History

4.4.5 UC4.S5: Administrator Abuse

This scenario models a situation where an administrator abuses his or her elevated privilege in an attempt to perform malicious actions under a new user account. The malicious administrator creates a new user to prevent log entries from containing the insider's username.

4.4.5.1 UC4.S5.Step 1: Taxonomy Development

− *Tools:* In this scenario, the insider leverages administrator privileges and existing Windows functionality to execute the attack.

− *Vulnerability:* As an administrator, the insider has permission to create new user accounts. However, the design of the system allows a privileged individual to arbitrarily create user accounts without verification for a supervisor or other administrator.

− *Actions:* To achieve the malicious objective, the insider performs the following steps in order. First, the insider creates a new local administrator account, *Mallory*. The insider then logs off and logs into the workstation (not the domain) using the newly created local administrator account. Next, the insider attaches an external hard drive to the workstation using USB-over-network. The malicious insider then copies the targeted file, Payroll.xlsx, from the C drive to the external hard drive. The insider completes their actions by dismounting the external drive and logging off.

− *Target:* The malicious insider is targeting sensitive payroll information containing addresses, full names, spouse and family information, salaries, bank account numbers, social security numbers, and credit card information for all employees of the organization.

- *Unauthorized Result:* The result of the insider's attack against the organization is a breach of confidentiality and disclosure of payroll information.

- *Objective:* The malicious insider is motivated by seeking revenge against an organization by stealing personally identifiable information (PII) regarding the employees and providing it to a competitor or nation-state.

4.4.5.2 UC4.S5.Step 2: VMI Observables

Observation of users on the system can be performed using CMAT-V. CMAT-V lists all users on the system, including service accounts, as well as the SID and home path for each user. This action can also be observed using the Windows security log.

Detection of the external hard drive is done using the previously mentioned

`HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{53f563`**`07`** `-b6bf-11d0-94f2-00a0c91efb8b}` and

`HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{53f563`**`0d`** `-b6bf-11d0-94f2-00a0c91efb8b}` registry entries. These two registry entries maintain a list of storage devices on the system.

The copying and pasting of Payroll.xlsx is observed through brute force string search because file clipboard operations cannot currently be detected via CMAT-V.

Table 4.14: UC4.S3 VMI Observables.

	Insider Action Description	VMI Observable
1	Create User Mallory	CMAT-V User List
2	User Login	None Identified
3	Connect USB Hard Drive	Registry Entry
4	Copy Payroll.xlsx to External Drive	Clipboard
5	Dismount External Drive	None Identified

4.5 UC5: Remote Access

Referring back to insider characteristics discussed in Chapter 2, another technique employed by malicious insiders is remote access. Using remote access allows the insider to perform their attack while not being distracted by coworkers or their currently assigned work task. Additionally, coworkers cannot observe any potentially malicious activity on the insider's screen and report the actions to a security manager within the organization.

4.5.1 UC5.S1: Workstation Remote Access

This scenario is representative of a user who uses Microsoft Remote Desktop Protocol (RDP) to access their workstation remotely, such as from their personal computer at home. The malicious insider uses RDP to steal data remotely from their work computer to a personal computer at home. RDP can be configured to use transport layer security (TLS) to prevent an organization from performing a man-in-the-middle (MITM) attack to determine the user's activity, thus defeating any network level defenses.

4.5.1.1 UC5.S1.Step 1: Taxonomy Development

- *Tools:* The malicious insider uses several tools to perform this attack. The insider's personal computer at home and their computer at work are both tools leveraged in this attack. On these two workstations, the insider only uses legitimate Windows commands.

- *Vulnerability:* The malicious insider is not exploiting any vulnerabilities within the organization. Even a correctly configured RDP session would enable the malicious insider to execute this attack. An organization's policy

may allow employees to work remotely and this is often seen as a benefit by prospective employees.

- *Actions:* The following actions are performed in order by the malicious insider to accomplish the objective of this scenario, theft of sensitive information. First, the malicious insider connects to their workstation via RDP. After successfully authenticating, the insider copies the desired documents. Specifically, the malicious insider first uses ctrl c and ctrl v to copy and paste FirewallSource.zip from C:\Users\tgreen\desktop to the C drive on his home computer. Next, the insider copies JointStrikeFighter.docx from W:\Projects\ to the C drive on his computer. The insider completes the scenario by disconnecting the RDP session.

- *Target:* The malicious insider targets several sensitive documents during this attack. The first, the firewall source code, is a project the insider is paid to work on, so accessing this should not raise suspicion. The second target is a document pertaining to the Joint Strike Fighter the organization is working on. These data are sensitive to the organization and would be financially damaging if a competitor obtained this information.

- *Unauthorized Result:* A successful attack by the insider results in disclosure of confidential information belonging not only to the organization, but to the government and possibly additional business partners.

- *Objective:* A malicious insider may have a variety of motives for performing this action; likely the insider is attempting to steal the information to sell it to a competitor or nation-state.

4.5.1.2 UC5.S1.Step 2: VMI Observables

The first action performed by the insider is connecting to their workstation via RDP. This action can be detected using an identified registry entry, `HKEY_CURRENT_USER\Volatile Environment\1`. This registry entry contains information regarding the user's current session on a workstation. When a user is locally connected, the value `SESSIONNAME` will be `Console` and `CLIENTNAME` will be `NULL`. If there are no current users logged on to a system, this key does not exist in the registry. Additionally,

`HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{28d78fad-5a12-11d1-ae5b-0000f803a8c2}\##?#Root#RDPBUS#0000#{28d78fad-5a12-11d1-ae5b-0000f803a8c2}\#TS001`, also records RDP session information. This registry key is not found in the registry unless an RDP session is currently in progress to the current workstation.

File copy and paste operations are both limited in detection due to CMAT-V limitations. These are both detected via string search through the full memory captures.

Table 4.15: UC5.S1 VMI Observables

Insider Action Description	VMI Observable
1	Connect to Workstation1 from Remote Machine via RDP	Registry Entries
2	Copy documents from remote machine to local machine	Clipboard

4.5.2 UC5.S2: Server Remote Access

This scenario is representative of a malicious insider who uses RDP to access one of the organization's servers. Users may need to remote into one of the servers for a variety of reasons, such as making a configuration change or accessing files only available on that system. As previously mentioned, RDP can be encrypted to prevent MITM attacks, which also defeats any network level traffic monitoring. Additionally, using RDP to a server could allow an insider to bypass security mechanisms on their workstation.

4.5.2.1 UC5.S2.Step 1: Taxonomy Development

- *Tools:* The insider uses tools similar to the previous remote access scenario to execute this attack. The insider's personal computer and the organization's server are both relied up for this attack. In contrast to the previous scenario, the attacker also has a piece of malware. On these two Windows computers, the insider only issues Windows commands.

- *Vulnerability:* Similar to the previous scenario, no vulnerabilities in configuration, implementation, or design are exploited during this scenario. It could be argued that the server is incorrectly configured to allow users to RDP to it, but administration of said server would be difficult using only command line.

- *Actions:* The following actions are performed in order by the malicious insider to accomplish the objective of this scenario, implanting malware on the insider's workstation for later execution. First, the malicious insider connects

the server MS01, which is the organization's mail and file server. After successfully authenticating, the mounts workstation1\Users\tgreen to a network drive. Workstation1 is the malicious insider's workstation. Next, the malicious insider copies DarkCometRAT.exe (DarkComet Remote Administration Tool), from their personal computer to ms01\tgreen\Desktop. After copying it to the desktop of the server, the malicious insider uses Ctrl C and Ctrl V to copy and paste the RAT from the server's desktop to the mounted network drive, workstation1. The insider completes the scenario by disconnecting the RDP session.

- *Target:* The malicious insider targets several sensitive documents during this attack. The first, the firewall source code, is a project the insider is paid to work on, so accessing this should not raise suspicion. The second target is a document pertaining to the Joint Strike Fighter the organization is working on. These data are sensitive to the organization and would be financially damaging if a competitor obtained this information.

- *Unauthorized Result:* As a result of the insider's actions, the insider placed malware on a workstation and can use it to increase access to the computer network and then steal, or corrupt information within the organization's network.

- *Objective:* The malicious insider's purpose for the attack is to put malware onto their workstation in preparation for an attack against the organization.

Windows Server 2003 records RDP information in the registry slightly differently than the aforementioned Windows 7 registry entries. The ControlSet001 registry entry is `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{28d78fad-5a12-11d1-ae5b-0000f803a8c2}\##?#Root#RDPDR#0000#{28d78fad-5a12-11d1-ae5b-0000f803a8c2}\#TS001` in Windows XP and Server 2003 and the Volatile Environment\1 registry entry is `HKEY_CURRENT_USER\Volatile Environment`.

Observation of the mounted network drive by the insider on the server (MS01) to the workstation (workstation1) is performed by the previously mentioned Map Network Drive MRU registry key.

The clipboard file operation steps are performed via brute force string search due to limitations within CMAT-V.

The final step, disconnecting the RDP session is performed using the aforesaid Volatile Environment and ControlSet001 registry entries.

Table 4.16: UC5.S2 VMI Observables.

	Insider Action Description	VMI Observable
1	Connect to MS01 via RDP	Registry Entry
2	Map network drive	Registry Entry
3	Copy DarkCometRAT.exe from personal computer to MS01	Clipboard
4	Copy DarkCometRAT.exe from MS01 to mapped network drive	Clipboard
5	Disconnect RDP session	Registry Entry

4.6 UC6: Clipboard Activity

The Windows clipboard is used frequently by users on a system for normal computer tasks. However, it can also contains valuable information regarding an insider attack and therefore examination of the Windows clipboard for post-incident investigation is extremely valuable in determining actions performed by the user [46].

Applying this principle to live introspection can significantly reduce the time between incident and detection and potentially generate real-time detection of malicious activity.

4.6.1 UC6.S1: Document Contents Copy and Paste

Copying and pasting between two documents is a common use of the Windows clipboard functionality. This scenario is representative of clipboard activity by a malicious insider who accesses an unauthorized document and copies and pastes the contents to a new document. The insider knows the organization works on a UAV for the Air Force, but is not familiar with where the documents are stored and therefore must search for the information.

4.6.1.1 UC6.S1.Step 1: Taxonomy Development

− *Tools:* To perform the attack, the malicious scenario uses only existing Windows user commands.

− *Vulnerability:* The insider is exploiting a configuration vulnerability within the organization which enables them to access all files on the network drive.

− *Actions:* The malicious insider performs the following actions in the order listed to conduct the attack. First, the insider opens Windows explorer and navigates to the network drive (10.1.0.205). Unlike several other scenarios, the insider does not map a network drive to this location. After accessing the network drive, the insider uses the Windows 7 search functionality to search for "UAV". The insider then opens AirForceBriefing.docx and selects all of the text. Ctrl C is used to copy the contents of the document to the clipboard. A new Word document is created by the malicious insider and named

UAVData.docx. The insider finishes the attack by pasting the contents of the clipboard and saving the document.

- *Target:* The target for the insider's attack is a classified briefing document the organization is going to present to the Air Force.

- *Unauthorized Result:* The effect of the insider's attack is creation of an unauthorized copy of UAV information. The insider may disclose this information to a third party, but that is outside the scope of this scenario.

- *Objective:* The goal of the attack is to obtain sensitive information regarding the organization's upcoming project.

4.6.1.2 UC6.S1.Step 2: VMI Observables

Several observables are identified to monitor for changes during this scenario in order to detect the malicious insider behavior. To observe the changes performed by the insider's first action, navigating to the network share, the registry key `HKEY_USERS\<SID>\Software\Microsoft\Windows\CurrentVersion\Explorer\Typ edPaths` is monitored, where SID is the Windows security identifier assigned to the a user. This registry key contains a list of the twenty-five most recently typed addresses into the Windows Explorer address bar. It is important to note the difference in information recorded between this and the aforementioned TypedURLs registry entry.

As previously described and shown, the registry entry WordWheelQuery is an MRUList containing the most recent one hundred Windows search queries. This is monitored to detect the insider's second action, searching for "UAV".

Monitoring the insider's file access activity is performed using the aforementioned …\Word\File MRU registry entry.

Lastly, the Word text clipboard operations are observed using the Windows clipboard functionality.

Table 4.17: UC6.S1 VMI Observables.

	Insider Action Description	VMI Observable
1	Navigate to network drive	Registry Entry
2	Search for "UAV"	Registry Entry
3	Open AirForceBriefing.docx	Registry Entry
4	Copy document contents	Clipboard
5	Create new Word document called UAVData.docx	Registry Entry
6	Paste document contents	Clipboard

4.6.2 UC6.S2: Document Contents and Web Browser Copy and Paste

Similar to the previous scenario, copying and pasting between a document and a web form is another common use of Windows clipboard. This scenario models an insider who uses an anonymous web form to exfiltrate information from the organization. The insider employs the Windows clipboard and Internet Explorer to perform the attack.

4.6.2.1 UC6.S2.Step 1: Taxonomy Development

− *Tools:* The malicious insider uses the Windows clipboard and Internet Explorer to perform this attack.

− *Vulnerability:* The insider does not exploit any configuration, implementation, or design vulnerabilities within the system. All of the actions performed by the insider are legitimate system commands.

- *Actions:* The following actions are performed in order by the malicious insider during this scenario. First, the insider accesses *unmanned systems icd draft v2-2 (aroc approved).docx* a document not related to the project the insider works on. Next, the insider opens Internet Explorer navigates to www.pastebin.com. The insider then uses Ctrl C and Ctrl V to copy and paste the contents of the document to Pastebin and submit the pasted contents.

- *Target:* The target for the insider's attack is an unclassified document containing detailed specifications of an upcoming UAV project.

- *Unauthorized Result:* The consequence of the insider's attack is public distribution of sensitive information. The organization has not publically released this information and doing so allows individuals to view it who have not been approved by the organization.

- *Objective:* The malicious insider's objective is to publicly release information regarding the organization's UAV program which will damage the organization financially and create a negative public image.

4.6.2.2 UC6.S2.Step 2: VMI Observables

The previously identified Word\FileMRU is used for observing the insider's file access activity. This registry entry maintains a list of recently accessed Microsoft Word files.

To observe browser activity, the TypedURLs registry entry and string scan of full memory capture are monitored. The full memory scan generates some false positives due

to a limited hexadecimal pattern matching technique, but it also captures any links contained on a page, such as links on a search engine.

Lastly, the clipboard is used to detect the text copy and paste operation. Text can currently be observed, but due to implementation limitations, source and destination file paths must be searched via full memory string search.

Table 4.18: UC6.S2 VMI Observables.

	Insider Action Description	VMI Observable
1	Open document not related to job	Registry Entry
2	Navigate to pastebin.com	TypedURLs
3	Copy and paste document contents to pastebin.com	Clipboard

4.6.3 UC6.S3: Outlook Email Contents and Web Browser Copy and Paste

The final malicious clipboard scenario performed is similar to scenario 0, with the key difference being the source application used for the text clipboard operation. A malicious insider

Similar to the previous scenario, copying and pasting between a document and a web form is another common use of Windows clipboard. This scenario models an insider who uses an anonymous web form to exfiltrate information from the organization. The insider employs the Windows clipboard and Internet Explorer to perform the attack.

4.6.3.1 UC6.S3.Step 1: Taxonomy Development

− *Tools:* In this modeled attack, the malicious insider uses Window 7 clipboard capacity, Microsoft Outlook 2007, and Internet Explorer 8 to accomplish the objective.

105

– *Vulnerability:* The insider does not exploit any design, configuration, or implementation vulnerabilities within Windows. It could be argued that a configuration error of a network level traffic monitor allows the user to visit pastebin.com, but the insider could use one of the many similar sites or create their own.

– *Actions:* To accurately model an attack, the listed actions are performed by the malicious insider in the order given. First, the malicious insider opens Internet Explorer and Outlook. The insider then double clicks an email containing sensitive text in the body of the email to open it in a new window. After opening the email, the insider copies all of the text to the clipboard. The insider directly navigates to pastebin.com and pastes the contents of the email.

– *Target:* The insider is targeting an email containing sensitive performance data about the organization's UAV program.

– *Unauthorized Result:* After performing this attack, the insider disseminates valuable information to an unlimited number of third-parties who are not authorized to possess this information.

– *Objective:* The malicious insider's motives are to seek revenge against the organization and damage any customers who have purchased the product by revealing limitations of the product.

A hexadecimal search is developed for obtaining email contents from the workstation's memory and is listed below. It is suspected that the string only detects HTML based emails and not plain-text emails.

3C68746D6C20786D6C6E733A763D2275726E3A736368656D61732D6D6963726F7
36F66742D636F6D3A766D6C2220786D6C6E733A6F3D2275726E3A736368656D61
732D6D6963726F736F66742D636F6D3A6F66666963653A6F666669636522

Observation of navigation to pastebin.com is performed using the hex search of a full memory capture and TypedURLs registry entry.

The last insider action is observed using CMAT-V's clipboard monitoring capabilities. As previously stated, detection of source and destination application is limited.

Table 4.19: UC6.S3 VMI Observables

	Insider Action Description	VMI Observable
1	Open Outlook	Running Process
2	Copy email contents	Hexadecimal Pattern / Clipboard
3	Navigate to pastebin.com	Registry Entry
4	Paste email contents	Clipboard

4.7 Summary

This chapter presented the malicious insider use cases using the methodology detailed in Chapter 3. It elaborated on each of the methodology steps, describing the motivation for each use case, described specific scenarios performed for each use case, broke each scenario down using the modified computer and network incident taxonomy, and enumerated VMI observables for detection of each action within the scenario.

V. Insider Threat Detection and Data Validation

To support the goal of this research, each scenario must be tested to determine if each observable identified can be monitored for possible insider activity. If a change is detected, an alert should be written for a security analyst to investigate further to determine if the user has malicious intent. Furthermore, to ensure previously identified observables only generate an alert for insider threat actions, the observables are tested with two different data sets not containing a malicious insider. One dataset is from manually generated normal user scenarios and the second dataset is from the Advanced Cyber Education (ACE) Hackfest containing computer network operations (CNO) actions by users.

This chapter focuses on addressing the detection of the malicious insider and validating the detection method for each scenario described in Chapter 4. Section 5.1 addresses printer use cases. Section 5.2 covers disabling defense tools. Section 5.3 discusses the successfulness of removable media detection. Section 5.4 focuses on suspicious employee behavior. Section 5.5 addresses remote access attack vectors and detection. Finally, Section 5.6 provides detection for clipboard scenarios. The chapter concludes with a summary in Section 5.7.

5.1 UC1: Printing Activity

Printers are frequently used by malicious and non-malicious insiders. Although printers are able to record information about print jobs, they cannot see into a user's workstation to determine how the user obtained the information and if they are authorized

to access it. The ability to observe an entire print job on a user's workstation enables an organization to more rapidly identify potentially malicious behavior.

5.1.1 UC1.S1.Step 3: Malicious Insider Detection

Examination of the generated alert reveals one new entry to the Driver3 registry entry. The output of this alert is shown in Figure 5.1. As mentioned in Chapter 3, all workstations in the experiment are connected to a Lexmark C782 network printer, so one known print driver exists in this key. Additionally, several other standard drivers exist on the system before performing this scenario. These drivers are the Microsoft XML Paper Specification (XPS), the Microsoft shared fax driver and Microsoft OneNote Driver. The OneNote driver is installed when Microsoft Office 2007 is installed. An organization could determine this alert alone is enough to cause serious suspicion of a user, if this user does not have a legitimate reason for having a personal printer.

```
Registry: \REGISTRY\MACHINE\SYSTEM
Key name: Version-3
Last updated: 2012-03-21 19:04:57

Subkeys:
            HP LaserJet 4350 PCL 5
            Lexmark C782 PS (MS)
            Microsoft Shared Fax Driver
            Microsoft XPS Document Writer
            Send To Microsoft OneNote Driver
```

Figure 5.1: UC1.S1 Alert – Driver3.

The next action by the malicious insider is mapping a network drive to their workstation. Depending on what drive and folder is mapped, an organization could also identify this single action as malicious if the drive or folder is outside of the user's work scope. However, in this scenario, the network folder accessed by the insider is within their work scope and he stores information on the network drive on a regular basis.

109

Figure 5.2 shows the resulting alert generated for the Map Network Drive MRU registry entry.

```
                    "Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: Map Network Drive MRU
Last updated: 2012-03-21 18:58:56

Subkeys:

Values:
REG_SZ          a               : \\10.1.0.205\Organization\Projects\Firewall
```

Figure 5.2: UC1.S1 Alert - Map Network Drive MRU.

Successful detection of the insider opening the targeted documented involves monitoring the previously identified Word\File MRU registry entry. Observing the change in this entry reveals malicious insider has opened Firewall Project Proposal.docx from the drive mapped to Z (determined previously to be 10.1.0.205) within the past ten minutes, shown in Figure 5.3.

```
                    "Software\Microsoft\Office\12.0\Word\File MRU"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: File MRU
Last updated: 2012-03-21 18:59:56

Subkeys:

Values:
REG_SZ          Item 1          : [F00000000][T01CD0794BBD4EBA0]*Z:\Firewall Project Proposal.docx
REG_SZ          Item 2          : [F00000000][T01CD078804E42660]*C:\Users\tgreen\Desktop\UAVData.docx
REG SZ          Item 3          : [F00000000][T01CD0787EDEAA650]*\\10.1.0.205\Organization\Classified
```

Figure 5.3: UC1.S1 Alert – Word\File MRU

The last action performed by the insider in this scenario is printing the Word document to the local printer. During analysis, no registry entries were observed to determine if a document is printed. Instead, the pattern identified in the VMI Observables

section is employed to capture the print job. Successful detection of the print job generates an alert and is shown in Figure 5.4.
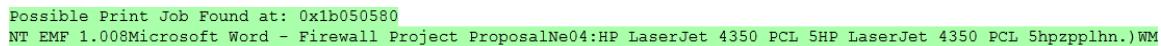
```
Possible Print Job Found at: 0x1b050580
NT EMF 1.008Microsoft Word - Firewall Project ProposalNe04:HP LaserJet 4350 PCL 5HP LaserJet 4350 PCL 5hpzpplhn.)WM
```

Figure 5.4: UC1.S1 Detection – Print Job

Through analysis of each step in this scenario, an organization can effectively employ a strategy to mitigate malicious insiders who use local printers to exfiltrate sensitive information. Each step in this malicious scenario is successfully alerted. As mentioned previously, an organization could alert on specific actions, but the combination of all steps is definitely malicious. Additional analysis by an organization enables completion of the malicious insider taxonomy to determine what the target and motivation is for the insider in this scenario.

5.1.2 UC1.S1.Step 4: Data Validation

Data validation for the Word File MRU registry entry and print jobs is contained in section 5.1.4 UC1.S2.Step 4: Data Validation.

Within the MIN data set, six alerts are generated; two on workstation4 and four on workstation5. All of these generated alerts are false positives and are a result of the user logging on/off from the system and causing the registry entries to be removed or added, triggering the alert generation. Examining the ACE Hackfest data, fourteen alerts are generated for Version-3 registry entry. All alerts except for two appear to be the result of users logging on and off from the system. Two alerts show the connection and disconnection of a Lexmark C534 printer from the computer BSOD-10.

111

Three alerts are generated for the Map Network Drive MRU registry entry in the MIN data set. Two of the alerts are by users accessing their respective folders within the organization's network drive. The third alert appears to be suspicious. User lscarlet mapped a network drive on her workstation to the CEO's performance review folder on the network drive. Additional investigation and accounting for lscarlet's job function reveals she is the employee relations advisor and therefore this information is related to her job duties. In the ACE data set, no alerts are generated for the registry entry.

Examination of both the …\Standard TCP/IP Port\Ports and …\Printers registry entries reveals they contain the same information. For non-malicious scenarios, four false positive alerts are generated. For the ACE Hackfest dataset, twelve alerts are generated and all are false positives.

5.1.3 UC1.S2.Step 3: Malicious Insider Detection

Detection of the first action with an observable is not successful. In a previous scenario, the insider had navigated to \\10.1.0.205 and therefore it is not added to the TypedPaths registry entry. Additionally, the timestamp on the registry entry is not updated.

Detecting a user's file search queries is valuable for signaling potential malicious actions by the user; searches outside of locations the user has access to or trigging on blacklisted terms can be the precursor to data exfiltration or destruction. Detection of the search performed by the user is successful, as shown in Figure 5.5. An alert is generated and the changed item is highlighted.

```
                "Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: WordWheelQuery
Last updated: 2012-03-21 19:38:52

Subkeys:

Values:
REG_BINARY    MRUListEx       :
0x00000000  01 00 00 00 00 00 00 00 ff ff ff ff            ............
REG_BINARY    0               :
0x00000000  55 00 41 00 56 00 00 00                        U.A.V...
REG_BINARY    1               :
0x00000000  43 00 6c 00 61 00 73 00 73 00 69 00 66 00 69 00  C.l.a.s.s.i.f.i.
0x00000010  65 00 64 00 00 00                              e.d...
```

Figure 5.5: UC1.S2 Alert - WordWheelQuery

Analysis of the alert discloses one initial value of "UAV" (the result of a different scenario) and an additional entry of "Classified". The malicious insider did not appear to have detailed knowledge of what information to look for; instead the insider appears to target the most sensitive information within the organization. Additionally, the Last Updated time changed between both entries, although the time does not match the time the action is performed.

Before execution of this scenario the targeted document is *Item 3* in the MRU list, meaning it is the least recently opened file on the system. After performing this scenario, *AirForceBriefing.docx* is the first item in the MRU list, indicating it is the most recently accessed Word document by the user, as illustrated in Figure 5.6. An additional entry, White-2011.docx, also exists in the MRU list, but this is from a different scenario.

```
                "Software\Microsoft\Office\12.0\Word\File MRU"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: File MRU
Last updated: 2012-03-21 19:39:18

Subkeys:

Values:
REG_SZ        Item 1       : [F00000000][T01CD079A4E5231E0]*\\10.1.0.205\Organization\Classified Research\UAV\AirForceBriefing.docx
REG_SZ        Item 2       : [F00000000][T01CD0799E4205F90]*C:\Users\tgreen\Desktop\White-2011.docx
REG_SZ        Item 3       : [F00000000][T01CD0794BBD4EBA0]*Z:\Firewall Project Proposal.docx
REG_SZ        Item 4       : [F00000000][T01CD078804E42660]*C:\Users\tgreen\Desktop\UAVData.docx
```

Figure 5.6: UC1.S2 Alert – Word\File MRU.

The final action performed by the malicious insider is printing the classified document to a local printer. It is determined that detection for this action from a hypervisor perspective would remain the same if the insider used a network printer instead of a local printer; with a network printer, an additional network connection may be observable and assist a security analyst in confirming the print job. The output of the alert showing the print job is in Figure 5.7.

```
Possible Print Job Found at: 0x1b050580
NT EMF 1.008Microsoft Word - AirForceBriefingNe04:HP LaserJet 4350 PCL 5HP LaserJet 4350 PCL 5hpz
```

Figure 5.7: UC1.S2 Alert – Print Job.

Although no time information is provide from analysis of the memory, an estimated time can be obtained comparing the two memory capture file names. The two file names, *w7_1_20120321142917_.dmp* and *w7_1_20120321145121_.dmp* indicate the print job occurred on 21 March 2012 between 14:29:17 and 15:51:21. Unfortunately, a weakness of this detection method is also observed during analysis of this action. Specifically, the print job performed in this action has the exact same memory address as a print job from a previous scenario. As a result, it may be possible for a malicious insider to hide their print job this action's detection method by printing several

114

unauthorized documents followed immediately by a legitimate document. It was not evaluated if a print job to a different printer would also store the information at the same memory address.

5.1.4 UC1.S2.Step 4: Data Validation

Analysis of the TypedPaths registry entry produced thirty-one alerts for the MIN network users and of these, eight are the result of users typing an address into the address bar; the remaining are a result of the registry entry being deleted/created when a user logs off/on to the workstation. Within the ACE Hackfest data, nine alerts are generated for the TypedPaths registry and of these, zero are the result of a user typing an address into Windows Explorer.

For the MIN data, five alerts are generated for WordWheelQuery registry entry. All searches except for one appeared to be benign and related to the user's position within the organization. The user on workstation2 performed a search for "Bath Salt" at approximately 14:20, which is not related to the user's work position and may be related to a substance abuse problem within the user's personal life. Examination of ACE data produced seven alerts for WordWheelQuery. All Windows Explorer searches did not appear to be malicious.

On the MIN network, twelve alerts are generated for the Word\File MRU registry entry. Examining each alert reveals the user's are operating within their work scope. For example, the CEO accesses and creates performance review documents. None of the generated alerts indicate suspicious behavior.

ACE Hackfest data produced nine alerts for Word\File MRU registry entry. Without context for each user's work scope, it is not possible to determine any

inappropriate or malicious file access. Nevertheless, the observed Word files accessed during the exercise include: Part 2 Problem 1.docx, pictures.docx, MemoTemplate.doc, MPFM0552.doc. A quick observation of the file names does not reveal any suspicious names, but the file could be obfuscated by the insider.

The final validation point is for print job alerts. In the MIN dataset, twenty-five alerts are generated with fifty-six percent coming from workstation3 and the remaining alerts coming from workstations 2, 4, and 5. Of these alerts, seven are a result of actual print jobs, the remaining are false positives. Workstation2 showed two print jobs, both to the network printer and both relating to the user's job position, which is consistent with the script. Workstation3 reported three print jobs, which again matches the script. Examination of the print jobs reveals none of the print jobs are related to the user's job function and two indicate suspicious personal behavior. The two suspicious print jobs suggest the user is involved in a local swinger's organization. Within the ACE dataset, no alerts are generated for print jobs. This is consistent with knowledge of the exercise; printing was not performed from workstations during the exercise.

5.1.5 UC1.S3.Step 3: Malicious Insider Detection

Analysis of the domain controller event logs reveals the user's successful logon at 00:12:41 on 22 March 2012, shown in Figure 5.8. Furthermore, it reveals the user is attempting to logon to their workstation, an authorized action. Although this action may be authorized, a sudden change in computer activity may signal potential malicious behavior.

```
Service Ticket Request:
User Name:              tgreen@THESIS.COM
User Domain:            THESIS.COM
Service Name:           WORKSTATION1$
Service ID:             S-1-5-21-956557069-3517617492-2542546096-1108
Ticket Options:         0x40810000
Ticket Encryption Type: 0x17
Client Address:         10.1.0.210
Failure Code:           -
Logon GUID:             {0ac87e67-d393-c7b8-5a31-a3d78b09e160}
Transited Services:     -
```

Figure 5.8: UC1.S3 Alert - Event Log

An alert is generated for the aforementioned observable Volatile Environment\1. The resulting alert, shown in Figure 5.9, displays the lasted updated time for the registry entry is significantly later than normal business activity. A discrepancy is observed between this value and the actual time on the insider's workstation. Both times are four hours later than the actual time the scenario is performed, but this is consistent throughout the insider experiment.

```
              Original                              Potentially Malicious
f 1                                         f 1
  2               "Volatile Environment\1"    2               "Volatile Environment\1"
  3                                           3
  4 Registry: \??\C:\Users\tgreen\ntuser.dat  4 Registry: \??\C:\Users\tgreen\ntuser.dat
  5 Key name: 1                               5 Key name: 1
t 6 Last updated: 2012-03-21 16:05:53       t 6 Last updated: 2012-03-22 04:00:27
  7                                           7
  8 Subkeys:                                  8 Subkeys:
  9                                           9
 10 Values:                                  10 Values:
 11 REG_SZ      SESSIONNAME    : Console     11 REG_SZ      SESSIONNAME    : Console
```

Figure 5.9: UC1.S3 Alert – Volatile Environment\1.

Detection of the insider mapping a network drive to *\\MS01\Organization* is successful. An alert is generated, signaling a change in the MRU list and potential

malicious activity on the workstation. The MRUList is modified to indicate the most recently added entry is the targeted network drive, as demonstrated in Figure 5.10.

```
                    "Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: Map Network Drive MRU
Last updated: 2012-03-22 04:02:29

Subkeys:

Values:
REG_SZ        a              : \\10.1.0.205\Organization\Projects\Firewall
REG_SZ        MRUList        : dcba
REG_SZ        b              : \\10.1.0.205\Organization\Mustard\Performance Reviews
REG_SZ        c              : \\10.1.0.212\Users\lscarlet
REG_SZ        d              : \\ms01\Organization
```

Figure 5.10: UC1.S3 Alert – Map Network Drive MRU.

After mapping the network drive, the insider then copies the five targeted documents to his/her local desktop from the network drive. Performing a brute force string search through the memory capture does reveal the five files are accessed by the insider.

The final action, printing the documents is detected with somewhat limited success. Examination of the RecentDocs MRUListEx alert reveals a significant change, as demonstrated in Figure 5.11. After performing this action, the value of MRUListEx significantly changes, adding 6 new entries to the beginning, in front of **0x0E**, the reference to Scarlet Files.lnk. Additionally, the subkeys beneath *...\RecentDocs* has a new entry, .xlsx. Xlsx is the extension used for Microsoft Excel documents, so examining the referenced pointed to by MRUListEx should contain at least one Excel document. Following the order found in MRUListEx provides detection of all five files, shown in Table 5..

118

```
                     "Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: RecentDocs
Last updated: 2012-03-22 04:04:24

Subkeys:
        .cpp
        .docx
        .iso
        .jpg
        .xlsx
        Folder

Values:
REG_BINARY    MRUListEx      :
0x00000000  12 00 00 00 11 00 00 00 0b 00 00 00 10 00 00 00    ................
0x00000010  0f 00 00 00 01 00 00 00 0e 00 00 00 0d 00 00 00    ................
0x00000020  07 00 00 00 0c 00 00 00 03 00 00 00 0a 00 00 00    ................
0x00000030  09 00 00 00 02 00 00 00 08 00 00 00 06 00 00 00    ................
0x00000040  05 00 00 00 04 00 00 00 00 00 00 00 ff ff ff ff    ................
```

Figure 5.11: UC1.S3 Alert – RecentDocs.

Table 5.1: UC1.S3 Alert – RecentDocs MRUList.

MRUListEx Value	Registry Value
0x12	Logger.cpp
0x11	PacketInspection.cpp
0x0B	AutoUpdate.cpp
0x10	VM Configuration.xlsx
0x0F	Passwords.xlsx
0x01	System and Security.lnk

The five targeted files by the malicious insider are all detected within the
RecentDocs registry key. Analysis reveals the three .cpp files are part of the project the
insider is working on, but the two .xlsx files are related to the network infrastructure of
the organization and outside of the work scope of the insider. Observation of the file
names reveals this information may be very damaging to the organization as it can expose
weaknesses regarding workstation configuration and possibly contain user account
passwords.

119

The previously employed analysis technique of scanning the memory dump for the print job pattern confirmed the theorized shortcoming discussed. That is, print jobs appear to use the same memory address and subsequent jobs overwrite previous entries. Before execution of this scenario, the previous print job of *AirForceBriefing.docx* remained in memory at address 0x1b050580. A nearby address of 0x1b050728 has the current print job written to it, overwriting part of the previous print job, shown in Figure 5.12.

```
Possible Print Job Found at: 0x1b050728
NT EMF 1.008Logger - NotepadDOT4_001HP LaserJet 4350 PCL 5HP LaserJet 4350 PCL 5hpzpp
```

Figure 5.12: UC1.S3 Alert - Print Job 1.

```
Possible Print Job Found at: 0x3591fa40
NT EMF 1.008tgreenAutoUpdate - Notepadtgreen\\WORKSTATION1HP LaserJet 4350 PCL 5HP LaserJet 4350 PCL 5C
```

Figure 5.13: UC1.S3 Alert - Print Job 2.

After printing all five documents, only two documents, Logger.cpp and AutoUpdate.cpp, are successfully detected. The malicious insider could mitigate successful detection of this action by printing several benign documents to the printer after Logger.cpp, such as an email or a weather report.

Through analysis of this scenario, the malicious insider's actions can be detected with somewhat limited success. The limitation of this analysis is the document printing; it cannot be determined that the malicious insider printed all of the target documents. Additional analysis of the spoolsv.exe process, the Windows print spooler service, did not show a handle from the process to the printed files. Nevertheless, it can be determined that the malicious insider accessed five files within and outside of their work

120

scope, opened all of them and printed at least two; enough information to signal malicious intent.

5.1.6 UC1.S3.Step 4: Data Validation

Within the MIN data set, thirty-four alerts are generated for the RecentDocs registry entry. Of these alerts, twenty-seven were the result of user actions and seven were false-positives. Some of the files accessed by the non-malicious users include: otf2.pdf, NewHire.docx, numtest.xlsx, BraxtonPuggle.jpg, Payroll.xlsx, JointStrikeFighter.docx, draft copy of fighter design.docx, Swingers May Invite.txt, Payroll notes.docx, and many non-work related dog, car, and swingers pictures. This registry entry also records control panel sections and Windows Libraries. The MIN data set showed users accessing Hardware and Sound, Network and Internet, the Documents Library, and My Pictures Library. Although some files do not appear to be work related, none of the recently accessed documents appear to be outside of each user's work scope.

Examination of the ACE data results in 10 alerts. Within the alerts, there are several Word, Excel, and several photos. However, several suspicious files are identified. The first is an html file with a file name over 45 characters in length and appearing to be random letters and numbers. Additionally, this file is located on CAE-02, which appears to be a Windows Server 2003 web server, suggesting the server may be compromised and serving a malicious file. The second suspicious file is named PsTools.zip, which appears to be the PsTools suite developed by Mark Russinovich. The alert is generated for a Windows 7 workstation, so it is extremely suspicious that a user's workstation would need to have a tool suite capable of executing remote processes, dumping event logs, and killing processes [72].

Data validation for Map Network Drive and print job memory scans are covered in sections 5.1.2 UC1.S1.Step 4: Data Validation and 5.1.4 UC1.S2.Step 4: Data Validation, respectively, and are not discussed in this section.

5.2 UC2: Disable Defense Tools

Defensive tools running on a user's workstation are the last line of defense for an organization for either an external attack targeting a workstation, or against a malicious insider. Tools running inside the guest can provide more information than introspection, but can also be subverted. These scenarios focus on identification of an insider disabling an organization's workstation defense tools.

5.2.1 UC2.S1.Step 3: Malicious Insider Detection

Examining the alert generated for disabling Microsoft Security Essentials, show in Figure Figure 5.14, reveals a new registry entry is created. This value indicates the antivirus has been disabled on the workstation, leaving it vulnerable to exploitation by the insider or an external attacker.

Figure 5.14: UC2.S1 Alert – Real-Time Protection

As this scenario only attempts to address if disabling the user-level system protection can be detected, additional actions are not performed by the malicious insider,

except re-enabling the antivirus. During the time that it is disabled, the malicious insider could run any public and detected malware tool without being prevented. The resulting registry change is displayed in Figure 5.15. A post-incident forensic analysis not involving Compiled Memory Analysis Tool – Virtual (CMAT-V) would not reveal the antivirus was disabled and re-enabled unless the event logs are also examined.

```
                          Potentially Malicious

              "Microsoft\Microsoft Antimalware\Real-Time Protection"

Registry: \SystemRoot\System32\Config\SOFTWARE
Key name: Real-Time Protection
Last updated: 2012-03-21 18:57:36

Subkeys:

Values:
REG_DWORD      DisableRealtimeMonitoring : 0
```

Figure 5.15: UC2.S1 Alert – Enable Antivirus

Detection for the malicious insider is declared successful as the change can be observed through VMI.

5.2.2 UC2.S1.Step 4: Data Validation

On the malicious insider network (MIN), only the insider threat's workstation generated potentially malicious alerts; non-malicious users who have Microsoft Security Essentials running on their computer did not generate an alert.

Analyzing the ACE Hackfest data, no instances of the DisableRealtimeMonitoring registry entry are observed. Several possible alerts were generated for differences in the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft

123

`Antimalware\Real-Time  Protection\DisableRealtimeMonitoring`
registry entry, but further analysis did not indicate potential malicious behavior.

These alerts are observed on computers `BSOD-8` and `BSOD-10` Windows 7 workstations. Further analysis of running processes from the exercise confirms Microsoft Security Essentials was running on the machines in question, as well as other machines during the exercise. Other machines had consistent registry entries throughout the exercise and had values identical to the *Potentially Malicious* portion of the alert for the aforementioned workstations. As a result, this alert can be declared non-malicious.

5.2.3 UC2.S2.Step 3: Malicious Insider Detection

As shown in Figure 5.16, the Security event log shows the time and user responsible for clearing the event log on the system. Detection of this scenario is extremely limited as there are no VMI observables and only one total observable. As a result, detection is determined to be successful, but limited due to lack of observables.

Figure 5.16: UC2.S2 Alert – EventLog

124

5.2.4 UC2.S2.Step 4: Data Validation

The malicious insider network did not contain any feature files with `mmc.exe` process running. Furthermore, analysis of event logs from the non-malicious workstations and servers did not contain any log clear events. Examination of the ACE Hackfest data did not reveal any instances of the `mmc.exe` process. Analysis of the log files did reveal several instances of team BSOD clearing the Security event log; however, these instances did not occur during the exercise and can be attributed to preparation for the exercise and therefore not-malicious. If the date of the exercise was not known, clearing the event log should be considered a malicious action.

5.2.5 UC2.S3.Step 3: Malicious Insider Detection

The first detectable action is the insider switching the browser to InPrivate browsing mode. The aforementioned string can be found in memory only while Internet Explorer is open and in InPrivate mode; after Internet Explorer is closed, no references to the aforementioned hex pattern are observed. Organizations should increase the frequency of full memory captures to accurately determine if InPrivate mode is in use. In addition to revealing the user is running Internet Explorer in private mode, it also reveals the titles for the web pages the user has browsed, although a separate solution is developed to determine sites visited. Figure 5.17 displays the detection of InPrivate browsing mode via the hex pattern VMI observable and also reveals web page titles can be determined by searching for this hex string; the insider visited a page called "Dark Comet RAT – official web site" while using InPrivate browsing.

125

```
InPrivate browsing found at: 0xee593de
r - [InPrivate]
kDarkComet RAT - official website - Windows

InPrivate browsing found at: 0xee596fc
Md
kpoison ivy hack - Google Search - Windows
```

Figure 5.17: UC2.S3 Alert – InPrivate Browsing History.

The next observable is the browsing history. As previously mentioned, several

hex patters are developed and a script is created to brute force the full memory captures

to find unique instances of possible URLs. The python script can be found in Appendix

G. The developed script also captures resources loaded on a webpage, even if the website

providing the resources is not directly accessed by the user. For example if the user visits

http://www.example.com and the page contains an image from *http://www.image.com*,

both will be reported by the script. Figure 5.18 illustrates several of the URLs found in

the generated alert.

```
Possible Browser History Found at: 0x1bbcc02
www.hackforums.net

Possible Browser History Found at: 0x1dbe0ea
www.ncbi.nlm.nih.gov

Possible Browser History Found at: 0x1dbe41a
www.poison-ivy.org
Possible Browser History Found at: 0x3f237ea
infiltrated.net

Possible Browser History Found at: 0x3f566b8
www.darkcomet-rat.com
```

Figure 5.18: UC2.S3 Alert – Overall Browsing History

For this scenario, the user explicitly visited *darkcomet-rat.com, poison-ivy.org,*

and *google.com.* Further examination of the alert reveals several other suspicious sites

126

that are referenced during the malicious insider's web browsing. These include infiltrated.net, evileyesoftware.com, nuclearwintercrew.com, and rootrulerz.com. Reproducing a Google search for "Poison Ivy RAT" reveals the several of the aforementioned suspicious sites to be links on the Google search result, as shown in Figure 5.19.

Shady **Rats** and **Poison Ivy**
infiltrated.net/index.php?option=com_content&view...id...
Written by J. Oquendo. Researchers at F-Secure were able to locate the initial message sent to EMC employees that ultimately lead to the RSA breach.

Download: Microsoft Malware Protection Center Threat Report ...
www.microsoft.com/download/en/details.aspx?id=27871
Nov 2, 2011 – Microsoft Malware Protection Center Threat Report - **Poison Ivy**. ...
Poison Ivy is a malware family whose primary objective is to allow complete ...

Pages similar to **www.poisonivy-rat.com**
Nuclear Winter Crew - NWC. RAT Remote administration tool ... - nuclearwintercrew.com
No-IP - No-IP is a dynamic dns provider (ddns), both free and paid, backed ... - no-ip.com
SweRAT - Forum, Topics, Replies, Last Post Info ... - swerat.com
Poison Ivy - Poison Ivy photos, photo quiz, FAQ, Stories, and a poster to ... - poison-ivy.org

Figure 5.19: UC2.S3 Alert – Google Search Reconstruction.

Although this could be argued to be a false positive by reporting more than the user's actual browser history, it is valuable to an analyst as it confirms the user's actions are indeed malicious. Examining the alert generated from the next memory dump, after completing the scenario, reveals all of the URLs written to memory in this scenario are no longer present in memory. An organization should consider this when determining the frequency of full memory captures.

The final action performed by the malicious insider in this scenario is downloading Poison Ivy RAT version 2.3.2. Similar to the previous detection method, scanning the full memory capture for a hex pattern is the only VMI observable. The

report generated immediately after executing the scenario encountered a control character and could not continue generating the alert. However, a subsequently generated alert reports the suspicious file, *PI2.3.2.rar*, as shown in Figure 5.20. A security analyst could take this information and quickly determine this is the same file name used for Poison Ivy RAT.
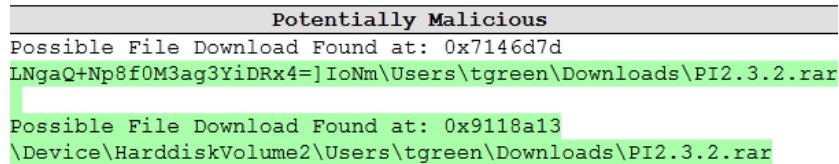
```
                   Potentially Malicious
Possible File Download Found at: 0x7146d7d
LNgaQ+Np8f0M3ag3YiDRx4=]IoNm\Users\tgreen\Downloads\PI2.3.2.rar

Possible File Download Found at: 0x9118a13
\Device\HarddiskVolume2\Users\tgreen\Downloads\PI2.3.2.rar
```

Figure 5.20: UC2.S3 Alert – File Download.

5.2.6 UC2.S3.Step 4: Data Validation

No alerts are generated for private browsing for other users within the non-malicious data set. Therefore, this detection method alerts to potentially malicious activity successfully for all analyzed data. For the ACE Hackfest network, no alerts are generated for InPrivate browsing.

Analysis of alerts generated for web browsing history reveals a significant number of false positives for both non-malicious and ACE Hackfest data. This can be partially attributed to freed memory having data written directly around the browser history pattern. To reduce the number of false positives, files alerts with a size less than one hundred kilobytes are omitted from analysis. Examining the alerts generated on the two servers on the non-malicious network reveal a maximum file size of fifty seven kilobytes. Both of these servers did not browse the internet during the non-malicious experiment, so they provide an estimated file size baseline.

128

Using the adjusted baseline file size, fifty five alerts are generated and of these, forty-six contain browsing history. Twenty-two alerts are generated for the ACE Hackfest data set. Of these alerts, eighteen contain actual browsing history.

During analysis of this scenario, a significant number of URLs are added while the insider is performing browsing the web to acquire a RAT. The alert file capturing this alert is three hundred forty two kilobytes and the subsequent alert, which shows the URLs being deleted, is three hundred thirty nine kilobytes. Therefore, using a minimum alert file size allows an organization to detect browsing history with greater confidence. To further aid detection, an organization should also consider employing blacklisting to alert for sites identified as especially dangerous. Since this method runs against a user's workstation memory, encryption techniques, such as SSH or VPN tunneling, employed by the insider do not subvert detection.

5.3 UC3: Removable Media

A common exfiltration method for malicious insiders is using removable media. Prohibiting removable media through the use of checkpoints and physical security is not effect as the data is no longer in the organization's control once it leaves the computer network. It is important for an organization to be able to monitor and detect suspicious removable media actions on a user's workstation to mitigate insider threats.

5.3.1 UC3.S1.Step 3: Malicious Insider Detection

Detection of the first action is successful using the registry key `53f56307-b6bf-11d0-94f2-00a0c91efb8b`. The change to this registry key is shown in Figure 5.21. The alert shows a Western Digital (WD) My Passport USBSTOR device is

connected to the workstation. Depending on an organization's policies, this single action may be determined to be malicious.

```
              "ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}"

Registry: \REGISTRY\MACHINE\SYSTEM
Key name: {53f56307-b6bf-11d0-94f2-00a0c91efb8b}
Last updated: 2012-03-21 20:23:03

Subkeys:
        ##?#IDE#DiskQEMU_HARDDISK_____0.8.2___#5&158eda0f&0&0.0.0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#USBSTOR#Disk&Ven_WD&Prod_My_Passport_0730&Rev_1016#57584D314135313837323230&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
```

Figure 5.21: UC3.S1 Alert – {53f56307-b6bf-11d0-94f2-00a0c91efb8b}.

The registry key `53f5630d-b6bf-11d0-94f2-00a0c91efb8b`, shown in Figure 5.22, also generates an alert for a new storage volume being added to the workstation. Analysis of a heavily used workstation revealed a difference between the two registry keys, but it is not clear why only certain devices are kept within the registry key `53f5630d-b6bf-11d0-94f2-00a0c91efb8b`, whereas the aforementioned key appears to contain a more extensive list. Nevertheless, detection of this action is successful.

```
              "ControlSet001\Control\DeviceClasses\{53F5630D-B6BF-11D0-94F2-00A0C91EFB8B}"

Registry: \REGISTRY\MACHINE\SYSTEM
Key name: {53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
Last updated: 2012-03-21 20:23:07

Subkeys:
        ##?#IDE#CdRomQEMU_QEMU_CD-ROM_____0.8.____#5&2d5296a&0&1.0.0#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#SCSI#CdRom&Ven_KernSafe&Prod_DVD-RAM_TM150&Rev_1.50#2&185905ec&0&000000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#Volume#{47005d14-4605-11e1-b702-806e6f6e6963}#0000000000100000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#Volume#{47005d14-4605-11e1-b702-806e6f6e6963}#0000000006500000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#VOLUME#{B70B51DE-7366-11E1-930A-0016360076D2}#0000000000100000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#VOLUMESNAPSHOT#HARDDISKVOLUMESNAPSHOT1#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#VOLUMESNAPSHOT#HARDDISKVOLUMESNAPSHOT2#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#VOLUMESNAPSHOT#HARDDISKVOLUMESNAPSHOT3#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
```

Figure 5.22: UC3.S1 Alert – {53f5630d-b6bf-11d0-94f2-00a0c91efb8b}.

Additionally, this list appears to allow correlation for copy and paste actions between volumes. As illustrated in Figure 5.23, the newest drive connected to the system is the third item in the list, a detail that will be important for detection of the second

action. However, no additional information could be obtained other than the obvious new entry was added. The aforementioned keys should be used as they provide details regarding what type of device was connected to the system.

```
                  "ControlSet001\Enum\Storage\Volume"

        Registry: \REGISTRY\MACHINE\SYSTEM
        Key name: Volume
        Last updated: 2012-03-21 20:23:05

        Subkeys:
                {47005d14-4605-11e1-b702-806e6f6e6963}#0000000000100000
                {47005d14-4605-11e1-b702-806e6f6e6963}#0000000006500000
                {b70b51de-7366-11e1-930a-0016360076d2}#0000000000100000
```

Figure 5.23: UC3.S1 Alert – Volume.

The second action performed by the malicious insider, copying and pasting the file *FirewallSource.zip* from the desktop to the external drive can be detected by monitoring the user's clipboard. As mentioned in the detection of the last action, it can be determined which storage volume the user copied the files to by examining the clipboard destination operation. Shown in Figure 5.24, the source for the copy operation is HarddiskVolume2 and tgreen's, the malicious insider, desktop. Although this copy action is within the work scope of the user, the subsequent paste operation is not and is suspicious.

```
0E3102A0   00 00 00 00 02 00 00 00 5C 00 44 00 65 00 76 00   ........\.D.e.v.
0E3102B0   69 00 63 00 65 00 5C 00 48 00 61 00 72 00 64 00   i.c.e.\.H.a.r.d.
0E3102C0   64 00 69 00 73 00 6B 00 56 00 6F 00 6C 00 75 00   d.i.s.k.V.o.l.u.
0E3102D0   6D 00 65 00 32 00 5C 00 55 00 73 00 65 00 72 00   m.e.2.\.U.s.e.r.
0E3102E0   73 00 5C 00 74 00 67 00 72 00 65 00 65 00 6E 00   s.\.t.g.r.e.e.n.
0E3102F0   5C 00 44 00 65 00 73 00 6B 00 74 00 6F 00 70 00   \.D.e.s.k.t.o.p.
0E310300   5C 00 66 00 69 00 6C 00 65 00 73 00 5C 00 46 00   \.f.i.l.e.s.\.F.
0E310310   69 00 72 00 65 00 77 00 61 00 6C 00 6C 00 53 00   i.r.e.w.a.l.l.S.
0E310320   6F 00 75 00 72 00 63 00 65 00 2E 00 7A 00 69 00   o.u.r.c.e...z.i.
0E310330   70 00 00 00 72 00 00 00 21 04 10 06 49 6F 4E 6D   p...r...!...IoNm
```

Figure 5.24: UC3.S1 Alert - Clipboard Source.

Figure 5.25 shows the resulting paste operation occurs to HarddiskVolume3. Recall from the previous action detection that the third connected volume is the external hard drive. The malicious insider has successfully copied the .zip file to an external drive and the last remaining step is to disconnect the removable media. As previously mentioned, no VMI observables were identified for this action.

```
035777F0   0C 00 00 00 00 00 00 00 79 63 02 06 00 00 00 80   ........yc.....€
03577800   5C 00 44 00 65 00 76 00 69 00 63 00 65 00 5C 00   \.D.e.v.i.c.e.\.
03577810   48 00 61 00 72 00 64 00 64 00 69 00 73 00 6B 00   H.a.r.d.d.i.s.k.
03577820   56 00 6F 00 6C 00 75 00 6D 00 65 00 33 00 5C 00   V.o.l.u.m.e.3.\.
03577830   46 00 69 00 72 00 65 00 77 00 61 00 6C 00 6C 00   F.i.r.e.w.a.l.l.
03577840   53 00 6F 00 75 00 72 00 63 00 65 00 2E 00 7A 00   S.o.u.r.c.e...z.
03577850   69 00 70 00 00 00 00 00 8D 62 02 06 00 00 00 80   i.p......b.....€
```

Figure 5.25: UC3.S1 Alert - Clipboard Destination.

5.3.2 UC3.S1.Step 4: Data Validation

Examining the non-malicious scenarios, three alerts are generated. Two of these alerts appear to have the same error several of the ACE workstations had. Specifically, the hard drive and CDROM device is not listed for several memory captures and causes an alert to be generated. However, examining all of the alerts revealed a true positive on workstation2. Further analysis of this alert reveals the device is a Western Digital

132

external hard drive. The registry key `...\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}` contains a subkey of: `##?#USBSTOR#Disk&Ven_WD&Prod_My_Passport_0730&Rev_1016#575848314533314C 4B543237&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}`. Per the non-malicious script, no information was copied to the drive, but nevertheless, an alert is successfully created for this potentially malicious activity by the user on workstation 2.

Executing the script to generate alerts for the ACE data resulted in 19 alert files being generated, indicating there is a change in one of the aforementioned registry keys for removable storage. Upon closer inspection, all of these are determined to be false-positives. In approximately half of the alerts, the alert reported a change for the registry keys

`HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{53f56307 -b6bf-11d0-94f2-00a0c91efb8b}`,

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Storage\Volume`, and `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{53f5630d -b6bf-11d0-94f2-00a0c91efb8b}`. The alerts showed the hard drive and CDROM devices missing from all three keys and being added in a subsequent memory capture. It is suspected that this occurred because either the guest had crashed or was in the process of restarting when a memory capture was performed resulting in the registry entry not being found. The remaining false-positives occurred because the last modified timestamp on `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Storage\Volume` changed between two memory captures. However, these alerts did not contain any additional changes and suggest the device may have been rebooted, but a dump did not occur during the reboot.

5.3.3 UC3.S2.Step 3: Malicious Insider Detection

The first generated alert is for a change in the Map Network Drive MRU registry entry. A third entry is added to this list and reveals the malicious insider connected to the computer 10.1.0.212 and to lscarlet's personal folder. Analysis of the alert generated for MountPoints2 confirms the detection of this action. An organization should be extremely suspicious of a user connecting to another user's workstation and accessing a personal directory.

```
                "Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: Map Network Drive MRU
Last updated: 2012-03-21 21:26:09

Subkeys:

Values:
REG_SZ          a                  : \\10.1.0.205\Organization\Projects\Firewall
REG_SZ          MRUList            : cba
REG_SZ          b                  : \\10.1.0.205\Organization\Mustard\Performance Reviews
REG_SZ          c                  : \\10.1.0.212\Users\lscarlet
```

Figure 5.26: UC3.S2 Alert – Map Network Drive MRU

Detection of the clipboard operations in this scenario is limited to brute force string searching due to CMAT-V limitations discussed in Chapter 3. Nevertheless, analysis of clipboard activity indicates the malicious insider copies NewHire, PayRoll and SocialSecurityNumber documents from lscarlet's folder to the local C drive. The malicious insider's actions

The final alert generated for the scenario is for the two CD Burning registry entries. As previously mentioned, analysis of these two entries revealed they appear to contain the same information when a disc is burned using existing Windows functionality. The highlighted items from the alert are only present during and

immediately after the burn process is complete. An alert is generated for the next memory

capture and shows the DefaultToMastered and Auto Close Wizard entries are removed.

```
                    "Software\Microsoft\Windows\CurrentVersion\Explorer\CD Burning"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: CD Burning
Last updated: 2012-03-21 21:28:10

Subkeys:
        Drives
        StagingInfo

Values:
REG_SZ          CD Recorder Drive : \\?\Volume{72d218fa-6ec3-11e1-8fc9-0016360076d2}\
REG_DWORD       DriveIndex      : 3
REG_DWORD       DefaultToMastered : 1
REG_DWORD       Auto Close Wizard : 0
```

Figure 5.27: UC3.S2 Alert - CD Burning

5.3.4 UC3.S2.Step 4: Data Validation

Within the non-malicious network, three CD Burn alerts are generated. One CD

Burning alert is generated for workstation 3 with user lscarlet, and another for bwhite on

workstation4. The third alert is a false positive. The alerts contains the same additions to

the CD Burning registry key as the malicious insider scenario, therefore it cannot be

determined if these actions are malicious by examining this one action. If an organization

explicitly prohibits optical media, this alert should be immediately acted upon. No alerts

are generated for the CD Burning registry entries within the ACE Hackfest data.

The non-malicious insider data set contains eleven alerts for MountPoints2

registry entry and of these, nine are the result of actual user actions. The remaining two

are false positives. None of these alerts appear to indicate insider actions. The ACE

dataset contains thirteen alerts for the MountPoints2 registry entry and of these thirteen

are false positives

135

5.4 UC4: Employee Behavior

The ability to detect a change in employee behavior can improve the response time an organization has to an insider attack. The following section discusses the success of detecting the employee behavior scenarios.

5.4.1 UC4.S1.Step 3: Malicious Insider Detection

Observing the alert generated for the network drive MRU reveals a new network drive is connected to the insider's workstation. Figure 5.28 shows the insider connected to the CEO's network folder and the MRUList is updated to indicate entry b is the most recently added entry to the key. This observed action is suspicious because the malicious insider is connecting to a folder not within their work scope.

```
REG_SZ        a               : \\10.1.0.205\Organization\Projects\Firewall
REG_SZ        MRUList         : ba
REG_SZ        b               : \\10.1.0.205\Organization\Mustard\Performance Reviews
```

Figure 5.28: UC4.S1 Alert - Map Network Drive

In addition to the network drive MRU, the aforementioned MountPoints2 registry key also shows the mounted drive. Figure 5.29 shows the examination of the alert generated for this registry entry. The Performance Reviews folder is a recently mapped network drive and added since the last full memory capture.

```
Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: MountPoints2
Last updated: 2012-03-21 19:36:07

Subkeys:
        ##10.1.0.205#Organization#Mustard#Performance Reviews
        ##10.1.0.205#Organization#Projects#Firewall
        CPC
        {47005d18-4605-11e1-b702-806e6f6e6963}
        {72d218fa-6ec3-11e1-8fc9-0016360076d2}
```

Figure 5.29: UC4.S1 Alert - Mountpoints2

Analysis of the Windows clipboard increases the evidence against the insider and reveals the insider copied six files (Crawford-2011.docx, Green-2011.docx, Green-2012.docx, Peacock-2011.docx, Scarlet-2011.docx, and White-2011.docx) from the Performance Review folder. In addition to accessing files outside of the insider's work scope, an organization should also be suspicious of such a large copy operation at one time from a non-local drive. Such a large copy and paste operation on a non-local drive could be deemed by an organization to be an unauthorized action.

```
1002F270   01 00 00 00 59 00 3A 00 5C 00 43 00 72 00 61 00   ....Y.:.\.C.r.a.
1002F280   77 00 66 00 6F 00 72 00 64 00 2D 00 32 00 30 00   w.f.o.r.d.-.2.0.
1002F290   31 00 31 00 2E 00 64 00 6F 00 63 00 78 00 00 00   1.1...d.o.c.x...
1002F2A0   59 00 3A 00 5C 00 47 00 72 00 65 00 65 00 6E 00   Y.:.\.G.r.e.e.n.
1002F2B0   2D 00 32 00 30 00 31 00 31 00 2E 00 64 00 6F 00   -.2.0.1.1...d.o.
1002F2C0   63 00 78 00 00 00 59 00 3A 00 5C 00 47 00 72 00   c.x...Y.:.\.G.r.
1002F2D0   65 00 65 00 6E 00 2D 00 32 00 30 00 31 00 32 00   e.e.n.-.2.0.1.2.
1002F2E0   2E 00 64 00 6F 00 63 00 78 00 00 00 59 00 3A 00   ..d.o.c.x...Y.:.
1002F2F0   5C 00 50 00 65 00 61 00 63 00 6F 00 63 00 6B 00   \.P.e.a.c.o.c.k.
1002F300   2D 00 32 00 30 00 31 00 31 00 2E 00 64 00 6F 00   -.2.0.1.1...d.o.
1002F310   63 00 78 00 00 00 59 00 3A 00 5C 00 53 00 63 00   c.x...Y.:.\.S.c.
1002F320   61 00 72 00 6C 00 65 00 74 00 2D 00 32 00 30 00   a.r.l.e.t.-.2.0.
1002F330   31 00 31 00 2E 00 64 00 6F 00 63 00 78 00 00 00   1.1...d.o.c.x...
1002F340   59 00 3A 00 5C 00 57 00 68 00 69 00 74 00 65 00   Y.:.\.W.h.i.t.e.
1002F350   2D 00 32 00 30 00 31 00 31 00 2E 00 64 00 6F 00   -.2.0.1.1...d.o.
1002F360   63 00 78 00 00 00 00 00 00 00 00 00 00 00 00 00   c.x.............
```

Figure 5.30: UC4.S1 Alert – Clipboard File Copy

The final action performed by the malicious insider is to open White-2011.docx. An alert is generated for the aforementioned Word MRU and upon examining this alert, the malicious insider's action is detected. Figure 5.31 shows the document as the second most recently accessed Word document; the first item in the list is related to a different scenario. Item 2 also reinforces the detection of the previous copy and paste action's detection since White-2011.docx is opened from tgreen's Desktop.

```
                    "Software\Microsoft\Office\12.0\Word\File MRU"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: File MRU
Last updated: 2012-03-21 19:39:18

Subkeys:

Values:
REG_SZ         Item 1          : [F00000000][T01CD079A4E5231E0]*\\10.1.0.205\Organization\Classified Res
REG_SZ         Item 2          : [F00000000][T01CD0799E4205F90]*C:\Users\tgreen\Desktop\White-2011.docx
REG_SZ         Item 3          : [F00000000][T01CD0794BBD4EBA0]*Z:\Firewall Project Proposal.docx
REG_SZ         Item 4          : [F00000000][T01CD078804E42660]*C:\Users\tgreen\Desktop\UAVData.docx
```

Figure 5.31: UC4.S1 Alert – Word\File MRU

5.4.2 UC4.S1.Step 4: Data Validation

5.1.2 UC1.S1.Step 4: Data Validation discussed data validation for network drive MRU and therefore it will not be repeated in this section. Additionally, validation for Word MRU is covered in section 5.1.2 UC1.S1.Step 4: Data Validation. Clipboard file copy validation is not possible due to CMAT-V limitations discussed in Chapter 3. During VMI observable analysis of the full captures for this scenario, no reliable hex pattern could be identified to assist with file copy and paste detection without generating an exorbitant amount of false positives.

5.4.3 UC4.S2.Step 3: Malicious Insider Detection

Detecting the iexplorer.exe process running is straightforward using CMAT-V's feature files. However, since this action is frequently performed by non-malicious users, an alert for potential malicious activity cannot be created. Examining the second action allows an alert to be generated. The alert generated for browsing history, as shown in Figure 5.32, reveals the insider visits truecrypt.org, the homepage for TrueCrypt.

```
Possible Browser History Found at: 0x21830d0
www.truecrypt.org

Possible Browser History Found at: 0x24d4a54
amch.questionmarket.com

Possible Browser History Found at: 0x27c523c
clk.atdmt.com

Possible Browser History Found at: 0x27c530a
view.atdmt.com
```

Figure 5.32: UC4.S2 Alert – Browser History.

Additional analysis on the generated alert also shows the download page (www.truecrypt.org/downloads) is also visited by the malicious insider. Examining the …\TypedURLs browser history provides additional evidence of the suspicious browsing action. Figure 5.33 shows the output of the alert generated for the TypedURLs registry entry; truecrypt.org is the most recently added item in the list.

```
                       Potentially Malicious

            "Software\Microsoft\Internet Explorer\TypedURLs"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: TypedURLs
Last updated: 2012-03-21 19:50:14

Subkeys:

Values:
REG_SZ        url1           :  http://www.truecrypt.org/
REG_SZ        url2           :  http://google.com/
REG_SZ        url3           :  http://ferrari.com/
REG_SZ        url4           :  http://ameritrade.com/
REG SZ        url5           :  http://go.microsoft.com/fwlink/?LinkId=69157
```

Figure 5.33: UC4.S2 Alert –TypedURLs.

Detection of the downloaded file is also successful via the memory dump scanning script. The alert correctly identifies TrueCrypt Setup 7.1a.exe as the downloaded file by the malicious insider in this scenario. Although the Poison Ivy

installation file is also listed in this alert, it is not highlighted as a new entry, meaning it is

contained in a previous alert, but the artifact remains in memory.

```
                         Potentially Malicious
Possible File Download Found at: 0x6ab9b85
\HarddiskVolume2\Users\tgreen\Downloads\TrueCrypt Setup 7.1a.exe

Possible File Download Found at: 0x7146d7d
nt.IE5\RP8A7QZU\1[1].jpg/IoNm\Users\tgreen\Downloads\PI2.3.2.rar

Possible File Download Found at: 0x868d9b5
\HarddiskVolume2\Users\tgreen\Downloads\TrueCrypt Setup 7.1a.exe

Possible File Download Found at: 0x9118a13
\Device\HarddiskVolume2\Users\tgreen\Downloads\PI2.3.2.rar
```

Figure 5.34: UC4.S2 Alert – File Download.

The TrueCrypt process is caught by CMAT-V's process feature file very quickly

after it is executed. The installation process and TrueCrypt process are both detected in

the process feature file. Table 5.2 contains a timeline of the installation and execution

process of TrueCrypt. Between the third and fourth entries, TrueCrypt.exe remains open;

however, it is omitted from the table for brevity. The first process name, TrueCrypt Setu,

is truncated by CMAT-V.

Table 5.2: UC4.S2 Detection – Running Process.

Process ID	Process Name	Feature File Timestamp
1676	TrueCrypt Setu*	03/21/2012 14:56:17
1728	TrueCrypt.exe	03/21/2012 14:58:29
1728	TrueCrypt.exe	03/21/2012 15:00:33
1728	TrueCrypt.exe	03/21/2012 15:13:52 (Last recorded entry)

* Denotes process name was truncated.

Mounting a TrueCrypt volume is successfully detected via the

…\MountedDevices registry entry. Figure 5.35 shows the alert generated from

…\MountedDevices registry entry. Of particular interest in this alert is line 119; the

140

TrueCrypt volume is assigned to the local disc letter E. This is important for verifying

any clipboard operations are sent to the E drive, which is unauthorized.

```
                         Potentially Malicious
 49
 50              "MountedDevices"
 51
 52Registry: \REGISTRY\MACHINE\SYSTEM
 53Key name: MountedDevices
 54Last updated: 2012-03-21 19:57:31
 55
 56Subkeys:
 57
 58Values:
 59REG_BINARY     \DosDevices\C:  :
108 0x000000a0  35 00 36 00 33 00 30 00 64 00 2d 00 62 00 36 00   5.6.3.0.d.-.b.6.
109 0x000000b0  62 00 66 00 2d 00 31 00 31 00 64 00 30 00 2d 00   b.f.-.1.1.d.0.-.
110 0x000000c0  39 00 34 00 66 00 32 00 2d 00 30 00 30 00 61 00   9.4.f.2.-.0.0.a.
111 0x000000d0  30 00 63 00 39 00 31 00 65 00 66 00 62 00 38 00   0.c.9.1.e.f.b.8.
112 0x000000e0  62 00 7d 00                                       b.}.
113REG_BINARY     \??\Volume{b70b51d3-7366-11e1-930a-0016360076d2} :
114 0x00000000  54 72 75 65 43 72 79 70 74 56 6f 6c 75 6d 65 58   TrueCryptVolumeX
115REG_BINARY     #{b70b51d8-7366-11e1-930a-0016360076d2} :
116 0x00000000  54 72 75 65 43 72 79 70 74 56 6f 6c 75 6d 65 58   TrueCryptVolumeX
117REG_BINARY     \??\Volume{b70b51d9-7366-11e1-930a-0016360076d2} :
118 0x00000000  54 72 75 65 43 72 79 70 74 56 6f 6c 75 6d 65 45   TrueCryptVolumeE
119REG_BINARY     \DosDevices\E:  :
120 0x00000000  54 72 75 65 43 72 79 70 74 56 6f 6c 75 6d 65 45   TrueCryptVolumeE
```

Figure 5.35: UC4.S2 Alert – MountedDevices.

Examining the Windows clipboard artifacts reveals the Word document is copied

from the insider's Z drive (mapped in another scenario to the organization's network

drive) to the E drive, the TrueCrypt volume. Detecting this action is the most important in

the scenario as it undeniably confirms malicious intent by the insider; the insider has

stolen company property.

The last action performed by the insider is to dismount the TrueCrypt encrypted

volume. Detection of this action is successful using the …\MountedDevices registry key

discussed in the VMI observables section and an alert is generated. As shown in Figure

5.36 the MountedDevices registry containing the value \DosDevices\E: has changed

to an unknown string. It is suspected that this string is a unique identifier assigned by Windows; however, analysis is not performed to determine the contents of the string because it is not necessary for detection of this action.

```
0x00000000   54 72 75 65 43 72 79 70 74 56 6f 6c 75 6d 65 58    TrueCryptVolumeX
REG_BINARY     #{b70b51d8-7366-11e1-930a-0016360076d2} :
0x00000000   54 72 75 65 43 72 79 70 74 56 6f 6c 75 6d 65 58    TrueCryptVolumeX
REG_BINARY     \??\Volume{b70b51d9-7366-11e1-930a-0016360076d2} :
0x00000000   54 72 75 65 43 72 79 70 74 56 6f 6c 75 6d 65 45    TrueCryptVolumeE
REG_BINARY     #{b70b51dc-7366-11e1-930a-0016360076d2} :
0x00000000   54 72 75 65 43 72 79 70 74 56 6f 6c 75 6d 65 45    TrueCryptVolumeE
```

Figure 5.36: UC4.S2 Alert - MountedDevices

5.4.4 UC4.S2.Step 4: Data Validation

Within MIN, the non-malicious users generated forty-seven file download alerts. A higher number is expected, as the non-malicious insiders explicitly downloaded files as part of the experiment. Due to the limited detection method in use for generating these alerts, only nineteen alerts are generated for file downloads.

Analysis of the twenty-three alerts generated during the ACE Hackfest reveals several very suspicious files downloaded by users during the exercise, as well as some benign downloads. Of these twenty three, only five contained actual file downloads. Several alerts contained multiple downloaded files. Suspicious files include: secretsaucecports.exe, psexec.exe, kasper_zaebal.exe, and 7z920.exe. Files of interest, but not necessarily malicious are: CAE-Scavenger_Part2-Q1.docx, Part2Question3.doc, Fall 2011 Mission Directive O-Plan ORIGINAL.docx, and Fall 2011 Mission Directive O-Plan (10 Aug 2011).docx.

Non-malicious users on the MIN experiment did not visit any potentially suspicious sites. Examination of the twenty-seven generated TypedURLs alerts reveals

these users visited primarily social networking, webmail, and sports sites. This aligns with the script followed by the non-malicious users. Of these twenty-seven alerts, five are false positives generated from users logging on/off from the system.

The TypedURLs registry entry generated thirteen alerts during the ACE Hackfest and of these five were false positives as a result of users logging on and off of the system. Most entries appear to be the result of normal web browsing; however the following suspicious entries are observed in the alert files: ftp://10.1.30.12/, Y:\nw4eiruow43hjrf89rn4q32n9w3480d983u9d843ud43jdc83w\1033\W3SVC1\87y3yq7 dn23y4nd2q73j4d87q4, peerblock (this appears to be a search using the address bar), and several web requests to 127.0.0.1.

Within the non-malicious data set, three alerts are generated for the MountedDevices registry entry. Two of the alerts show the connection of the KernSafe TotalMounter product on workstation2 and the third is potentially malicious, showing a device is mounted to the E: drive. Examination of the MountedDevices registry entry within the ACE dataset revealed ten alerts and all are false positives. The alerts indicate the user logged on and off of the system and as a result, it shows the adding and removal of the default CD-Rom device provided by Xen to the guest.

5.4.5 UC4.S3.Step 3: Malicious Insider Detection

Detection of this scenario is successful, except for one action performed by the insider. Figure 5.37 shows the successful detection of the command line actions performed by the insider.

```
        Command Line History:
*****************************************************
CommandProcess: conhost.exe Pid: 3236
CommandHistory: 0x300de0 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 7 LastAdded: 6 LastDisplayed: 6
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
Cmd #0 @ 0x2fe760: cd Desktop
Cmd #1 @ 0x2fe780: mkdir files
Cmd #2 @ 0x2fe7a0: cd files
Cmd #3 @ 0x2f41f0: cp "z:\FirewallSource.zip" c:\Users\tgreen\Desktop\files
Cmd #4 @ 0x2f4270: copy "z:\FirewallSource.zip" c:\Users\tgreen\Desktop\files
Cmd #5 @ 0x2fe7c0: ftp --help
Cmd #6 @ 0x305140: ftp martincrawford.net
```

Figure 5.37: UC4.S3 Alert – Command Line History

Although the file copy is successfully detected and reported in the alert, ftp is an external program and therefore command line history analysis does not reveal the actual commands issued once connected to the ftp server. Despite not fully capturing all ftp commands, examining the alert generated reveals very suspicious behavior by the insider; copying files and then immediately performing an ftp to an unauthorized site should cause an organization to increase monitoring of the user or take immediate action.

5.4.6 UC4.S3.Step 4: Data Validation

Examining the non-malicious MIN data, two alerts are generated. Analysis of these alerts shows the user on workstation3 issues the command *ipconfig*. This command shows TCP/IP configuration information for the current computer. Two alerts are generated because the first alert shows a change in command history between memory captures at 16:22 and 16:32. The second alert is generated because cmd.exe is closed after memory dump 16:32 and before the next capture causing the command history to be lost.

144

Two alerts are generated for the ACE Hackfest data for command line usage. The first alert is a false positive; it does not contain any valid command line commands. The second generated alert shows the user executed the *ipconfig dir, netstat*, and *ping* commands. These commands could be considered suspicious, but without knowledge of the user's job position, it cannot be determined if these are commands they would normally execute or a precursor to an attack.

5.4.7 UC4.S4.Step 3: Malicious Insider Detection

Unlike the previous scenario, detection for file deletion is much more straightforward because an external program is not called from the command line. The malicious insider issues the *del* command individually on files, allowing the resulting alert to contain all of the deleted filenames. Workstation3 is mounted as a network drive in a previous scenario. Figure 5.38 shows the output from the generated alert for this scenario, detailing the actions performed by the malicious insider.

```
        Command Line History:
**************************************************
CommandProcess: conhost.exe Pid: 2568
CommandHistory: 0x320de0 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 6 LastAdded: 5 LastDisplayed: 5
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
Cmd #0 @ 0x3162d0: net use
Cmd #1 @ 0x31f100: x:
Cmd #2 @ 0x31f130: dir
Cmd #3 @ 0x326138: del funnypic.jpg
Cmd #4 @ 0x321080: del german_shepherd_dog_664_12.jpg
Cmd #5 @ 0x316c80: del SocialSecurityNumbers.docx
Cmd #6 @ 0xff539a89: ???
```

Figure 5.38: UC4.S4 Alert – Command Line History.

5.4.8 UC4.S4.Step 4: Data Validation

Data validation for this scenario is identical to the previous scenario since both rely heavily on command line history. As previously described, two alerts are generated

for both the ACE and MIN data sets. The MIN alerts are determined to be benign. The true positive ACE alerts cannot be evaluated without knowledge of the user's profession within the organization.

5.4.9 UC4.S5.Step 3: Malicious Insider Detection

The CMAT-V user list successfully reports the additional user added to the system. The output of *Mallory S-1-5-21-3020999182-1602362634-1125454158-1001 C:\Users\Mallory* from CMAT-V should immediately draw suspicion to the workstation to determine who created a new user account. Consulting the Windows event logs reveals tgreen, the malicious insider, created the new account on the workstation1 domain, which indicates the account is a local account.


```
A user account was created.

Subject:
        Security ID:            S-1-5-21-956557069-3517617492-2542546096-1115
        Account Name:           tgreen
        Account Domain:         THESIS
        Logon ID:               0x1dd42

New Account:
        Security ID:            S-1-5-21-3020999182-1602362634-1125454158-1001
        Account Name:           Malory
        Account Domain:         WORKSTATION1
```

Figure 5.39: UC4.S5 Alert – Event Log.

An alert is generated for the malicious insider's action of connecting the external hard drive. Figure 5.40 shows the resulting alert. From the alert, it can be determined the malicious insider connected a Western Digital My Passport brand drive to the system.

146

```
                    "ControlSet001\Control\DeviceClasses\{53F5630D-B6BF-11D0-94F2-00A0C91EFB8B}"

Registry: \REGISTRY\MACHINE\SYSTEM
Key name: {53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
Last updated: 2012-03-22 04:01:44

Subkeys:
        ##?#IDE#CdRomQEMU_QEMU_CD-ROM_____0.8.____#5&2d5296a&0&1.0.0#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#SCSI#CdRom&Ven_KernSafe&Prod_DVD-RAM_TM150&Rev_1.50#2&185905ec&0&000000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_&PROD_PATRIOT_MEMORY&REV_PMAP#07A207000018888D&0#{53F56307-B6BF-11D0-94F2-00A0C
        ##?#STORAGE#Volume#{47005d14-4605-11e1-b702-806e6f6e6963}#0000000000100000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#Volume#{47005d14-4605-11e1-b702-806e6f6e6963}#0000000006500000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#VOLUME#{B70B51DE-7366-11E1-930A-0016360076D2}#0000000000100000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#VOLUME#{B70B5238-7366-11E1-930A-0016360076D2}#0000000000100000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#VOLUMESNAPSHOT#HARDDISKVOLUMESNAPSHOT1#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#VOLUMESNAPSHOT#HARDDISKVOLUMESNAPSHOT2#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
        ##?#STORAGE#VOLUMESNAPSHOT#HARDDISKVOLUMESNAPSHOT3#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
```

Figure 5.40: UC4.S5 Alert – {53f5630d-b6bf-11d0-94f2-00a0c91efb8b}

5.4.10 UC4.S5.Step 4: Data Validation

Investigation of the MIN data did not reveal any new accounts created during the exercise. Examining the ACE Hackfest data shows two accounts are created by the BSOD team, but before the exercise started. Both accounts are created on station1 and the accounts are named Test and Frank. Team CAE also created several local accounts before the start of the exercise; these accounts are: FTP, Work11, _vmware_user_, CAE, Work6, workstation5, user, and workstation1. Detection of removable devices in ACE and MIN data sets is discussed in section 5.3.2 UC3.S1.Step 4: Data Validation.

5.5 UC5: Remote Access

An insider employing remote access allows the attack to avoid detection by a coworker. Additionally, it also allows the insider threat to concentrate on only the attack and not try to pretend to be working on a task assigned by their supervisor. The ability of an organization to detect malicious remote access improves their insider threat mitigation strategy.

5.5.1 UC5.S1.Step 3: Malicious Insider Detection

An alert is successfully generated for the first action by the insider, connecting via RDP. A security analyst should also be suspicious of the time that the insider is connecting to the workstation; examining the time the alert is generated reveals the time the key is changed is at 0:20. Figure 5.41 shows the output of the alert generated for this action. Note how session name has changed to RDP-Tcp#0, indicating the current connection to the workstation is not via Console (local), but over RDP. Clientname has changed from null to displaying the hostname of the insider's home computer.

```
            "Volatile Environment\1"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: 1
Last updated: 2012-03-22 04:12:47

Subkeys:

Values:
REG_SZ          SESSIONNAME     : RDP-Tcp#0
REG_SZ          CLIENTNAME      : AUTOPWN
```

Figure 5.41: UC5.S1 Alert – Volatile Environment\1.

To further confirm the alert generated for the Volatile Environment registry key, the ControlSet001 registry entry is created and as a result, an alert is generated. Figure 5.42 shows the changed entries added to this registry key; analysis reveals the Base Name is TS, which is an abbreviation for Terminal Server, indicating the insider's workstation at work is serving the RDP session to the insider's personal computer at home. Port Description also confirms the hostname of the insider's personal computer at home.

```
                         "ControlSet001\Control\DeviceClasses\{28d7

Registry: \REGISTRY\MACHINE\SYSTEM
Key name: Device Parameters
Last updated: 2012-03-22 04:12:48

Subkeys:

Values:
REG_DWORD      Port Number      : 1
REG_SZ         Base Name        : TS
REG_SZ         Port Description : AUTOPWN:   PRN4
REG_DWORD      MaxBufferSize    : 0
REG_SZ         Client Device Name : \;PRN4:1\tsclient\PRN4
```

Figure 5.42: UC5.S1 Alert – Windows 7 #TS001

Examination of the Windows clipboard reveals the insider copies FirewallSource from C:\Users\tgreen\Desktop to the C drive on their home computer. Additionally, the Joint Strike Fighter documents are copied to the insider's home computer from a previously mounted network share.

5.5.2 UC5.S1.Step 4: Data Validation

Analysis of the ACE Hackfest data results in generation of seven alerts. Further examination reveals but all of these are a result of users logging off of their workstation and causing the Volatile Environment registry entry to be deleted; no non-malicious users performed an RDP connection from an external computer to their workstation. For the non-malicious users, nine alerts are generated, but these are also the result of users logging off of their workstation.

5.5.3 UC5.S2.Step 3: Malicious Insider Detection

Detection of the malicious insider's first action is successful and an alert is generated for each monitored registry entry. Examining the first portion of the alert generated, as shown in Figure 5.43, shows the Port Description changed since the last memory capture and also reveals the hostname for the connected computer.

```
                    "ControlSet001\Control\DeviceClasses\{28d78fad-5.

Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\system
Key name: Device Parameters
Last updated: 2012-03-22 04:22:25

Subkeys:

Values:
REG_DWORD       Port Number     : 1
REG_SZ          Base Name       : TS
REG_SZ          Port Description : AUTOPWN:  PRN4
REG_DWORD       MaxBufferSize   : 0
REG SZ          Client Device Name : \;PRN4:2\tsclient\PRN4
```

Figure 5.43: UC5.S2 Alert – Server 2003 #TS001

The alert also contains additional information regarding the RDP connection by the insider, contained in Figure 5.44. The Volatile Environment registry entry confirms the connected computer name is AUTOPWN, but also shows the security analyst the connecting user is tgreen and the user authenticated to the domain using ADS01 domain controller. The time of the attack can be determined by examining the timestamp on the alert; the attack occurs at approximately 00:30.

```
            "Volatile Environment"

Registry: \Device\HarddiskVolume1\Documents and Settings\tgreen\NTUSER.DAT
Key name: Volatile Environment
Last updated: 2012-03-22 04:22:25

Subkeys:

Values:
REG_SZ          LOGONSERVER     : \\ADS01
REG_SZ          USERDNSDOMAIN   : THESIS.COM
REG_SZ          CLIENTNAME      : AUTOPWN
REG_SZ          SESSIONNAME     : RDP-Tcp#2
REG_SZ          APPDATA         : C:\Documents and Settings\tgreen\Application Data
```

Figure 5.44: UC5.S2 Alert – Server 2003 Volatile Environment

The alert generated for Network Drive MRU shows a new network drive is connected to the server by the malicious insider. Figure 5.45 demonstrates the detection of this step is successful and the user connecting the network drive is tgreen.

```
                    "Software\Microsoft\Windows\CurrentVersion\Explorer\Map Ne
Registry: \Device\HarddiskVolume1\Documents and Settings\tgreen\NTUSER.DAT
Key name: Map Network Drive MRU
Last updated: 2012-03-22 04:26:36

Subkeys:

Values:
REG_SZ          a                : \\workstation1\Users\tgreen
REG_SZ          MRUList          : a
```

Figure 5.45: UC5.S2 Alert - Map Network Drive MRU

Examination of the clipboard contents reveals the malicious insider copied the executable from their personal computer to the server, MS01. The malicious insider first copied the file to the server's desktop and then to the mounted network drive, which is their workstation.

5.5.4 UC5.S2.Step 4: Data Validation

For non-malicious users, twelve alerts are generated. All alerts except for one are the result of normal logon and logoff activity by either users on their own workstation or network administrators managing the servers. One alert however reveals a user connected from workstation5 to MS01. Examination of the clipboard contents for workstation5 and MS01 at the same time this alert is generated does not reveal any suspicious files being copied and appears to be text from a Word document. Therefore this alert can be dismissed as a false positive. Eight are generated for a RDP during the ACE Hackfest data analysis. Examining the generated alerts reveals they are all a result of users logging on or off on their workstation and not using RDP to connect to another workstation.

5.6 UC6: Clipboard Activity

The last use case examined is clipboard activity. Due to CMAT-V limitations, file clipboard operations are pre-determined for malicious insider scenarios. This also limits

the ability to validate data for these scenarios against ACE Hackfest data as file clipboard operations cannot be examined.

5.6.1 UC6.S1.Step 3: Malicious Insider Detection

The malicious insider's first action is successfully alerted on and examining Figure 5.46 reveals the action. MS01 (10.1.0.205) is a public network drive, so accessing this through Windows explorer is not a suspicious action, but when combined with the following events, reveals malicious intent.

```
            "Software\Microsoft\Windows\CurrentVersion\Explorer\TypedPaths"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: TypedPaths
Last updated: 2012-03-21 17:32:48

Subkeys:

Values:
REG_SZ         url1              : \\10.1.0.205
```

Figure 5.46: UC6.S1 Alert – TypedPaths.

An analyst should become suspicious during examination of the WordWheelQuery alert, shown in Figure 5.47. The insider performs a search and the item searched for since the last memory capture is "UAV". Knowledge of the insider's profession allows this action to be suspicious since the insider works on the firewall project and not the UAV.

152

```
                 "Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: WordWheelQuery
Last updated: 2012-03-21 17:27:24

Subkeys:

Values:
REG_BINARY      MRUListEx      :
0x00000000  00 00 00 00 ff ff ff ff                          ........
REG_BINARY      0              :
0x00000000  55 00 41 00 56 00 00 00                          U.A.V...
```

Figure 5.47: UC6.S1 Alert – WordWheelQuery.

The third generated alert, contained in Figure 5.48, reveals the insider accessed a document titled *AirForceBriefing.docx*, an unclassified document relating to the organization's classified UAV project. Item 1 in the MRU list of Word documents is created by the insider to paste the contents of *AirForceBriefing.docx* into.

```
                 "Software\Microsoft\Office\12.0\Word\File MRU"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: File MRU
Last updated: 2012-03-21 17:28:31

Subkeys:

Values:
REG_SZ        Item 1      : [F00000000][T01CD078804E42660]*C:\Users\tgreen\Desktop\UAVData.docx
REG_SZ        Item 2      : [F00000000][T01CD0787EDEAA650]*\\10.1.0.205\Organization\Classified Re.
```

Figure 5.48: UC6.S1 Alert – Word\File MRU.

Examining the clipboard data contained within the memory capture revealed Microsoft Word contents, confirming the insider's intentions are malicious. Brute force searching the memory capture also confirmed the clipboard operation employed the two previously mentioned files.

153

5.6.2 UC6.S1.Step 4: Data Validation

Validation of the Word MRU list is covered in detail in section 5.1.2 UC1.S1.Step 4: Data Validation and therefore is not covered again in this section. Additionally, WordWheelQuery analysis for ACE and MIN data sets is covered in section 5.1.4 UC1.S2.Step 4: Data Validation. Analysis of the clipboard contents is limited due to inability to determine source and destination files or file directories.

5.6.3 UC6.S2.Step 3: Malicious Insider Detection

The insider's first action, accessing the UAV Word document, is successfully alerted through the File MRU registry entry. Figure 5.49 shows the resulting alert and the targeted document is the most recently accessed document in the MRU list.

```
            "Software\Microsoft\Office\12.0\Word\File MRU"

Registry: \??\C:\Users\tgreen\ntuser.dat
Key name: File MRU
Last updated: 2012-03-21 20:07:22

Subkeys:

Values:
REG_SZ        Item 1        : [F00000000][T01CD079E3A156400]*\\ms01\Organization\Classified Research\
REG_SZ        Item 2        : [F00000000][T01CD079A4E5231E0]*\\10.1.0.205\Organization\Classified Res
REG_SZ        Item 3        : [F00000000][T01CD0799E4205F90]*C:\Users\tgreen\Desktop\White-2011.docx
REG_SZ        Item 4        : [F00000000][T01CD0794BBD4EBA0]*Z:\Firewall Project Proposal.docx
REG_SZ        Item 5        : [F00000000][T01CD078804E42660]*C:\Users\tgreen\Desktop\UAVData.docx
```

Figure 5.49: UC6.S2 Alert – Word\File MRU

Continuing the attack, the insider opens Internet Explorer and navigates to pastebin.com. Detection of this action is successful using both the full memory capture scan (shown in X) and using the TypedURLs registry entry. If the malicious insider did not directly navigate to pastebin.com, the detection would still occur using the full memory capture.

154

```
Possible Browser History Found at: 0x20e1d8
pastebin.com

Possible Browser History Found at: 0x20eada
view.atdmt.com

Possible Browser History Found at: 0x20ed4c
clk.atdmt.com
```

Figure 5.50: UC6.S2 Alert – Browser History

Finalizing the attack, the malicious insider pastes the contents of the clipboard to the web browser. Examining the clipboard contents reveals Microsoft Word text. Brute force searching of the memory capture also reveals the clipboard text source is the aforementioned word document.

5.6.4 UC6.S2.Step 4: Data Validation

Difficulties exist in examination of the MIN non-malicious user data. Clipboard contents can be observed, but without the ability to determine the source or destination application for the copy and paste operation, only blacklist string searches can be performed. As mentioned in section 5.2.6 UC2.S3.Step 4: Data Validation, extensive browser history alerts are generated for the MIN users. Alerts generated for the TypedURLs entry, as examined in section 5.4.4 UC4.S2.Step 4: Data Validation, revealed the MIN users did not directly navigate to any suspicious sites.

Examining the ACE dataset for suspicious actions without knowledge of each user's job functions is difficult. It cannot easily be determined if a user has authorized access to a file by only knowing a filename. Furthermore, examining clipboard text contents is not useful, unless an organization employed a blacklisting technique to monitor for prohibited or suspicious terms.

5.6.5 UC6.S3.Step 3: Malicious Insider Detection

Observation of the user's email activity is successful. The resulting alert shows the insider emailing the project lead for specifications on the UAV project and the project lead replying with the information. Figure 5.51 shows the response and targeted information. Examining email server log confirms the contents of the email.

```
<p class=MsoNormal><span style='color:#1F497D'>Sir,<o:p></o:p></span></p>

<p class=MsoNormal><span style='color:#1F497D'>Top speed is 30, And Altitude 
is 60K.<o:p></o:p></span></p>

<p class=MsoNormal><span style='color:#1F497D'><o:p> </o:p></span></p>

<p class=MsoNormal><span style='color:#1F497D'>THIS INFORMATION IS CLASSIFIED.<o:p></o:p></span></p>
```

Figure 5.51: UC6.S3 Alert – Email Contents

Determining the browser history is slightly more limited than in previous scenarios. The TypedURLs registry entry timestamp is not updated when a previously typed URL is accessed again. Instead, only the full memory scan can successfully detect the insider's browsing activity.

```
Possible Browser History Found at: 0xa2146f
download.windowsupdate.com

Possible Browser History Found at: 0xaa001e
pastebin.com
```

Figure 5.52: UC6.S3 Alert - Browser History

Clipboard detection is not successful for this scenario. It is observed that the clipboard contains data, but it cannot be determined what the contents are. It is possible that the clipboard contents were overwritten between the insider copying the text to the clipboard and the next memory snapshot. This reinforces the importance of performing full memory captures on a short time interval.

5.6.6 UC6.S3.Step 4: Data Validation

The MIN data set produced one hundred twenty-eight alert files over all seven machines. Almost all alerts over twenty-five kilobytes in size contained an actual email. Some of the false positives are a result of Outlook being closed on a user's workstation and portions of the email body being overwritten in memory, while the hexadecimal identification pattern remained in memory, causing the system to believe new email contents existed within memory. An additional discovery is made while analyzing the MIN data set; the exchange server maintains all emails in memory. As a result, approximately twenty seven percent of alerts are generated by the Exchange server. Emails persisting in the Exchange server's memory could be useful to an organization if an incident occurred where an insider modified or deleted the Exchange logs which existed on disk. No alerts are generated for the active directory server, which is logical as it does not handle any of the email process.

Examining the data from the ACE Hackfest resulted in 18 generated alerts. Several of the generated alerts seemed to contain only binary information within the email body; perhaps these emails only contained an attachment. Other emails appeared to discuss tasks such as "problem 1", "question 9", "questions 1 and 8", reference to an "official memo", and other exercise related discussion. Since the detection method relies on pattern matching, it also detects several entries that have been dereferenced by Outlook and partially overwritten by the system, causing an alert to be incorrectly generated.

Analysis from the TypedURLs registry entry for the ACE and MIN data sets can be found in section 5.4.4 UC4.S2.Step 4: Data Validation.

5.7 Results

As stated in Chapter 3, the goal of this research was to determine if virtual machine introspection can be leveraged to alert to potential insider threats. This research does not use performance metrics, such as time to detection or detection accuracy, but only seeks to provide a solution for the research goal. Chapter 3 also introduces the methodology to decompose use cases into specific scenarios performed. Scenarios are then broken down using a modified computer and network incident taxonomy and to facilitate identification of VMI observables.

Table 5.3 summarizes the results of generated alerts for each previously identified observable. The first entry in a column indicates false positives, which could be eliminated by enhancing the functionality of the alert generation tool. An alert was generated because the observable changed significantly from the previous memory capture, but in these cases, it is often the result of logging on or off of a system. It is important to note that one logoff event would cause all listed observables to generate a false positive as the registry keys no longer exist. The second value, contained in parenthesis, is generated as the result of actual user action on a workstation or server. Examination of these alerts does not indicate malicious insider behavior. The values contained within the square brackets indicate potentially malicious insider behavior and should be investigated. For both data sets, this value is fairly limited. When looking at the false positives reported, it is important to note that the reported numbers are not taking into account additional steps performed in a scenario. These values represent only the analysis of each observable individually.

Table 5.3: False Positives in Observables.

ACE	Normal	VMI Observable
12(2)	6(0)	HKLM\SYSTEM\ControlSet001\Control\Print\Environments\ Windows NT x86\Drivers\Version-3
12(0)	4(0)	HKLM\SYSTEM\ControlSet001\Control\Print\Monitors\Stand ard TCP/IP Port\Ports
12(0)	4(0)	HKLM\SYSTEM\ControlSet001\Hardware Profiles\0001\System\CurrentControlSet\Control\Print\Printe rs
9(0)	23(8)	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer \TypedPaths
0(0)	0(2)[1]	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer \Map Network Drive MRU
0(9)	0(12)	HKCU\Software\Microsoft\Office\12.0\Word\File MRU
0(7)	0(5)	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer \WordWheelQuery
7(0)	9(0)	HKCU\Volatile Environment\1 & HKCU\Volatile Environment
5(5)	7(27)	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer \RecentDocs
0(0)	0(0)	HKLM \SOFTWARE\Microsoft\Microsoft Antimalware\Real-Time Protection\DisableRealtimeMonitoring
19(0)	2(0)[1]	HKLM\SYSTEM\ControlSet001\Control\DeviceClasses\{53f56 307-b6bf-11d0-94f2-00a0c91efb8b}
19(0)	2(0)[1]	HKLM\SYSTEM\ControlSet001\Control\DeviceClasses\{53f56 30d-b6bf-11d0-94f2-00a0c91efb8b}
19(0)	2(0)[1]	HKLM\SYSTEM\CurrentControlSet\Enum\Storage\Volume\
13(0)	2(9)	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer \MountPoints2
0(0)	1(0)[2]	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer \CD Burning
0(0)	1(0)[2]	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer \Shell Folders\CD Burning
5(4)[4]	5(27)	HKCU\Software\Microsoft\Internet Explorer\TypedURLs
10(0)	0(2)[1]	HKLM\SYSTEM\MountedDevices
8(0)	11(1)	… {28d78fad-5a12-11d1-ae5b-0000f803a8c2}\#TS001 (Both W2003 & W7)
0(0)	0(0)	InPrivate Browsing
0(0)	20(3)[2]	Print Jobs
18(5)	28(19)	File Downloads
4(18)	9(46)	Browsing History
172(50)[4]	**136(115)[11]**	**Totals**

Reducing false positives of the tool (the first entry in each column in Table 5.3) would not impact the success rate for alerting of the malicious insider scenarios. These alerts are the result of a registry entries no longer existing after a user logs off of the system, or conversely, the creation of a registry entry when a user logs onto the system.

Consequently, eliminating these false positive alerts would not impact the success of alerting for malicious insiders, but would affect the cost of investigation.

To eliminate the false positives as a result of user action (the second entry in each column in Table 5.3), significant enhancements would need to be made. Specifically, each user's work scope and related documents would need to be known by the alert generation tool, which would provide a white list for user actions. Unfortunately, this would require significant overhead to setup and maintain and would generate alerts any time a user created a new document. An alternative to this would be a blacklist on either per user or per group basis. It is expected that this would also suffer from a similar over head as whitelisting. A third solution would be to begin to incorporate behavioral analysis into the alert generation process.

Table 5.4 summarizes the results of comparing the developed alert methods against the malicious insider, non-malicious, and ACE Hackfest data sets. All generated malicious insider threat scenarios are successfully detected within the insider data set, when accounting for all assumptions previously identified. Specifically, all scenarios were performed within the time span of one memory capture and each action was performed in the order listed. For non-insider data sets, generated alerts are examined to determine if they match the same alerts generated for a malicious insider scenario. For example, an alert may be generated for a remote access action, but clipboard and file registry entry alerts are not generated. The ACE Hackfest could not have an outcome determined for the three scenarios associated with use case six (UC6) as the source and destination files or programs could not be determined for the clipboard operations. Without knowing source and destination, an analyst essentially only has a small piece of

160

text they could inspect for blacklisted strings. For non-malicious data, no insider activity

was alerted when examining the scenario as a whole.

Table 5.4: Malicious Insider Scenario Detection.

Scenario	Insider	Non-Malicious	ACE Hackfest
UC1.S1	Detected	Not Present	Not Present
UC1.S2	Detected	Not Present	Not Present
UC1.S3	Detected	Not Present	Not Present
UC2.S1	Detected	Not Present	Not Present
UC2.S2	Detected	Not Present	Not Present
UC2.S3	Detected	Not Present	Not Present
UC3.S1	Detected	Not Present	Not Present
UC3.S2	Detected	Not Present	Not Present
UC4.S1	Detected	Not Present	Not Present
UC4.S2	Detected	Not Present	Not Present
UC4.S3	Detected	Not Present	Not Present
UC4.S4	Detected	Not Present	Not Present
UC4.S5	Detected	Not Present	Not Present
UC5.S1	Detected	Not Present	Not Present
UC5.S2	Detected	Not Present	Not Present
UC6.S1	Detected	Not Present	Unknown
UC6.S2	Detected	Not Present	Unknown
UC6.S3	Detected	Not Present	Unknown

5.8 Summary

This chapter presented the six use cases previously identified in Chapter 4. It

presented steps three and four, detailing the detection for each observable action

previously identified and comparing the detection techniques against two data sets not

containing an insider threat.

VI. Conclusions and Recommendations

This chapter presents a summary of the insider threat alert mechanisms evaluated in Chapter 5. The first section discusses the importance of the research. Next, the second section discusses limitations identified within the research. The third section provides ideas for future research in the area of virtual machine introspection (VMI) to alert for potential malicious insiders. The chapter concludes with a discussion of conclusions drawn from the thesis.

6.1 Importance of Research

This research is significant in that it presented a reproducible methodology which can be employed to alert for additional malicious insider attack vectors. Additionally, it provided alert mechanisms for six use cases and their corresponding scenarios.

Furthermore, the research advanced mitigation techniques for the problem of insider threat. A novel approach to workstation insider threat alerting is presented and functions in a transparent manner to the individual under observation. Transparency to the user provides the insider with a potential false sense of security by not knowing of the existence of the organization's monitoring capabilities. Future work in this area, specifically the items mentioned in Section 6.3, which extends upon this thesis, would further aid mitigation strategies.

6.2 Limitations

The methodology presented in this research was successful in alerting for all insider threat scenarios. However, several scenarios had a limited number of available

observables, or relied solely upon event logs, making detection of a sequence of potentially malicious actions difficult.

6.3 Recommendations for Future Research

This research focused on leveraging existing functionality of Compiled Memory Analysis Tool – Virtual (CMAT-V) to alert for potentially malicious insider behavior within a guest virtual machine. Discussed below is future work recommendations based on information learned during the process of this research.

6.3.1 Clipboard Research

CMAT-V currently contains functionality to capture text from the clipboard. The clipboard is accessed from the kernel by traversing PsLoadedModuleList to find win32k.sys. After this is located, win32k.pdb is obtained, either locally or from the Microsoft symbol server. With this information, the symbol *gSharedInfo* can be located. With this information, CMAT loops through clipboard formats which are referenced by a pointer from Windows Station. After determining the clipboard format, the handle is converted and clipboard data obtained [45].

6.3.1.1 User Level Clipboard

The main problem with user level clipboard file operations is the lack of a global structure to maintain the clipboard value. For text operations, the function GetClipboardData uses an offset to *gSharedInfo* to maintain persistence of the clipboard text between applications. File operations are handled somewhat differently. During analysis, multiple functions were identified using both MSDN documentation and IDAPro which potentially contained useful clipboard file information.

The MSDN documentation lists a clipboard format named CF_HDROP. The data is a STGMEDIUM structure [73]. STGMEDIUM consists of a *tymed* double word which records the storage medium type, a union of seven variables and a pointer to allow the sending process to control how the data is released [74]. CF_HDROP was of interested because through the DROPFILES structure, a double null terminated character array that contained the source and destination file names and paths could be obtained. However, no functions with the necessary data structures which would have allowed recovered of the double null terminated array never triggered breakpoints when using WinDbg. These functions appear to be only called by applications and not by Windows itself.

The last investigated area was IDataObject. Described by [75], the process for an application developer to obtain file names from files on the clipboard is to first call OleGetClipboard to obtain the object's IDataObject interface. The IDataObject contained within OleGetClipboard contains a pointer to an unknown structure or function. MSDN documentation does not elaborate on what is contained within the structure [76]. It only lists several functions which can be called by application developers. Reverse engineering IDataObject without any documentation or disassembled code is difficult and due to the limited time constraints for this project, it was not completed and left for future work. To develop an alternative solution, the kernel level clipboard was examined.

6.3.1.2 Kernel Level Clipboard

Unlike the user level clipboard, obtaining information regarding clipboard command line file operations through kernel function calls is fairly straightforward. The Microsoft Developer Network (MSDN) site contains several documented functions for clipboard operations contained within Kernel32.dll. The desired information, source and

destination file path, are listed as input variables to these functions. Several additional functions were identified using IDAPro which appeared to possibly be relevant to the clipboard operation. Using WinDbg, breakpoints were placed on these functions and a Ctrl C and Ctrl V file copy and paste was performed. Unfortunately none of the breakpoints were hit. Using drag and drop, as well as right click copy and paste also produced the same results. The final file copy and paste technique performed was via command line *copy* function. This function did trigger several breakpoints. The most useful breakpoint appeared to be CopyFileEx. On the first call to this function, the source file path pointer is contained at EBP + 8h and the destination file path pointer is contained at EBP + Ch. The pointers to the file paths only exist within the CopyFileEx function. It is not possible to access this data after execution of the function completes. No functions within Kernel32 were observed to be called when a copy and pasted was performed in any other manner than command line.

6.3.1.3 Kernel File Operations

Within the kernel, several other potentially useful file operation functions are identified. The first is MoveFileExW. This function is called when a user performs a drag and drop move operation (but not a copy), or the command line *mov* operation. During the function execution, EBP + 8h contains a pointer to the source file. Similarly, EBP + Ch contains a pointer to the destination file for the move operation.

The final file operation observable through kernel functions is file deletion. Being able to detect every file deletion operation would greatly assist real-time insider threat detection on a user's workstation. The function DeleteFileWStub was determined via WinDbg to be called when a user performs a permanent file delete (Shift + Delete) or the

165

command line function *del*. Within this function, the pointer at EBP + 8h contains the full path to the file being deleted.

When a user performs a regular delete using only the delete key, the aforementioned MoveFileExW is called. This makes sense because technically the action performed is simply moving the document to the recycle bin. EBP + 8h again contains a pointer to the source document to be deleted. Following this pointer gives the full file path to the document. EBP + Ch also contains a pointer and following this reveals the destination for the delete operation is the recycle bin. However, the destination file path is not simply C:\$RECYCLE BIN. The destination path is C:\$RECYCLE BIN\SID\$(unknown).<EXTENSION>. In place of the filename, a string of seven characters appeared after the dollar sign. No investigation was performed to determine how the string was generated as the goal of detecting a file being deleted was accomplished.

6.3.2 Printer Research

As previously discussed, reverse engineering was performed to determine information regarding Windows print contents, but could not be implemented due to limitations with CMAT-V. Developing functionality to capture print queue contents would greatly improve introspection capabilities and eliminate the need for brute force string searches through memory and consequently reduce and likely eliminate false positives.

Analysis and reverse engineering of the Windows printer was only performed against a Windows 7 Service Pack 0 virtual machine; it is suspected that Windows 7

Service Pack 1, Windows Vista and Windows Server 2008 likely have similar printing behavior, but further analysis would need to be performed to verify.

Functions within User32 and WinSpool were evaluated to determine if any were suitable for capturing desired information regarding print job information. Analysis started with these functions because they were loaded when printing a sample Notepad document. However, the desired information could not be readily obtained and SpoolSV was selected for further investigation. Using IDAPro to disassemble the Windows binaries and WinDbg to debug a running SpoolSV, detailed information regarding print jobs was found in GetJob. Two GetJob functions exist within SpoolSV; GetJobW which handles Unicode data, and GetJobA which is used for ANSI. This function contains a pointer which references either a `JOB_INFO_1` or a `JOB_INFO_2` structure, both of which contain valuable information regarding print jobs.

Multiple test print jobs were sent while debugging the VM using WinDbg. GetJobW is called several times while the system is printing a single document. In order to obtain the `JOB_INFO` information, cbBuf was monitored until it contained a value. cbBuf (EBP + 18h) is an input variable to the GetJob function and contains the size of the array, represented in bytes. Additionally, after this occurs, the variable pJob may be null. According to Microsoft's documentation, cbBuf is sent with a null value to GetJob in order to determine the required buffer size for pJob. pJob (EBP + 14h) will then contain a pointer to either a `JOB_INFO_1` or a `JOB_INFO_2` structure. Following this pointer reveals the desired print job information. Specific items of interest include pPrinterName, pMachineName, pUserName, pDocument, and TotalPages. It was observed that PagesPrinted did not seem to change; it is suspected that this may be updated through a

different callstack. Within the GetJob function, hPrinter pointer did not reliably contain information regarding the printer and when data was in the pointer, it only contained the printer and not any information regarding the document; it was easier to allow SpoolSV to continue execution to obtain the JOB_INFO structure.

6.3.3 VMware and VirtualBox

During the final documentation stages of this thesis, development of a driver which enables VMware implementation of CMAT-V was completed. This driver executes within the virtual machine, so it could be compromised by malware or a technically proficient insider, but it is well suited for academia settings. VMware does not suffer from the same USB limitations as Xen. Additionally, the driver would allow file clipboard monitoring as well as monitoring the GetJob SpoolSV function. VMware does not have a publically available application programming interface (API) developers can access, meaning developing a solution that does not have any components executing of the guest would require extensive reverse engineering of the VMware application itself in addition to reverse engineering any desired Windows components.

VirtualBox, a virtualization package developed by Oracle has a publically available API, with several documented functions that appear to allow arbitrary reading and writing of guest memory. Unfortunately, these functions are listed as not implemented as of this writing (4.1.12), but are listed as possibly being implemented sometime in version 4. It may be advantageous to develop a proof-of-concept once these features are implemented within the VirtualBox API. If the API's features reveal similar information as XenAccess, CMAT-V should be ported to VirtualBox. Moving to VirtualBox would eliminate the reliance on XenAccess and its older dependencies and

168

would allow either Linux or Windows hosts, increasing the potential adoption of this research. Furthermore, it would also eliminate the need for extensive workarounds discussed in Chapter 3 for USB devices.

6.4 Summary

This research provided several benefits to the field of insider threat mitigation, by investigating the ability to signal potential malicious insider activity on a Windows workstation through the use of VMI. The first benefit provided is the solution to the initial research goal and demonstrating a novel approach.

The second benefit provided is indirectly provided through obtaining a solution to the research and required development of a repeatable methodology. This methodology enabled each use case to have scenarios generated to support several possible attack vectors. A finite methodology allows organizations employing this insider threat mitigation solution to rapidly analyze attack vectors specific to their network, or add additional vectors not covered in this research.

Appendix A: Acronym List

ACE	Advanced Cyber Education
AFIT	Air Force Institute of Technology
ASCII	American Standard Code for Information Interchange
AV	Anti-Virus
CCR	Center for Cyberspace Research
CD	Compact Disc
CD-RW	Compact Disc-Rewritable
CDX	Cyber Defense Exercise
CentOS	Community Enterprise Operating System
CERT	Computer Emergency Response Team
CIA	Central Intelligence Agency
CIA	Confidentiality, Integrity, Availability
CMAT-V	Compiled Memory Analysis Tool – Virtual
CNE	Computer Network Exploitation
CNO	Computer Network Operations
DSS	Defense Security Service
DoD	Department of Defense
DNS	Domain Name System
DPI	Deep Packet Inspection
DVD	Digital Video Disc
FTP	File Transfer Protocol
IC	Intelligence Community
IDS	Intrusion Detection System
IP	Intellectual Property
ISO	International Organization for Standardization
ITDS	Insider Threat Detection System
MAC	Media Access Control
MRU	Most Recently Used
NIC	Network Interface Card
NSA	National Security Agency
PDB	Program Database
RAID	Redundant Array of Independent Disks
RAT	Remote Administration Tool
RDC	Remote Desktop Connection
RDP	Remote Desktop Protocol
S	Scenario
SSD	Solid State Drive
SSH	Secure Shell
SSL	Secure Sockets Layer
UC	Use Case

VIX   Virtual Introspection tools developed for Xen
VLAN   Virtual Local Area Network
VMI   Virtual Machine Introspection
VMM   Virtual Machine Manager
WPAFB   Wright-Patterson Air Force Base

Appendix B: Xen Configuration

Parameter	Value on Host1	Value on Host2
host	insiderthreat	insiderthreat2
release	2.6.18-194.el5xen	2.6.18-194.el5xen
version	#1 SMP Fri Apr 2 16:16:54 EST 2010	#1 SMP Fri Apr 2 16:16:54 EST 2010
machine	i686	i686
nr_cpus	4	4
nr_nodes	1	1
sockets_per_node	2	1
cores_per_socket	2	2
threads_per_core	1	2
cpu_mhz	2992	2693
hw_caps	(omitted for brevity)	(omitted for brevity)
total_memory	4093	8149
free_memory	383	383
node_to_cpu	node0:0-3	node0:0-3
xen_major	3	3
xen_minor	1	1
xen_extra	.2-194.el5	.2-194.el5
xen_caps	xen-3.0-x86_32p hvm-3.0-x86_32 hvm-3.0-x86_32p	xen-3.0-x86_32p hvm-3.0-x86_32 hvm-3.0-x86_32p
xen_pagesize	4096	4096
platform_params	virt_start=0xf5800000	virt_start=0xf5800000
xen_changeset	unavailable	unavailable
cc_compiler	gcc version 4.1.2 20080704 (Red Hat 4.1.2-48)	gcc version 4.1.2 20080704 (Red Hat 4.1.2-48)
cc_compile_by	mockbuild	mockbuild
cc_compile_domain	centos.org	centos.org
cc_compile_date	Fri Apr 2 14:50:47 EDT 2010	Fri Apr 2 14:50:47 EDT 2010
xend_config_format	2	2

Appendix C: Malicious Insider Threat Script

Scenario	Start Time	Finish Time	Steps
UC2.S2	11:38	11:40	Open Windows Event Viewer Clear Security Log Clear System Log
Workstation times changed to pre-DST time (Now +1 of Dom0 time). Servers did not change.			
UC2.S1	12:57	12:59	Open Microsoft Security Essentials Disable Real-Time Protection
*	14:57	14:57	Open Microsoft Security Essentials Enable Real-Time Protection
UC2.S3	13:10	13:16	Open InPrivate browsing Attempt to download DarkCometRAT (Failed) Search for Poison Ivy Download PoisonIvyRAT
UC6.S1	13:25	13:32	Navigate to 10.1.0.205 Search for UAV Open AirForceBriefing.docx Copy text to clipboard Create new Word document named UAVData.docx Paste contents of document and save
UC1.S1	14:55	15:06	Map MS01\Organization\Project\Firewall to the Z drive. Connect printer to workstation Print FirewallProjectProposal
UC4.S1	15:35**	15:37	Map Y drive to \\10.1.0.205\Organization\Mustard\Performance Reviews Copy all listed performance reviews to Desktop Open White-2011.docx
UC1.S2	15:37	15:40	Navigate to \\10.1.0.205 Search for "Classified" Open AirForceBriefing.docx Print to local printer
UC4.S2	15:50	16:05	Open Internet Explorer and navigate to truecrypt.org Download Truecrypt and install with default settings Create and mount truecrypt volume as E: Copy FirewallProjectProposal.docx from network drive to TrueCrypt volume Dismount TrueCrypt volume
UC6.S2	16:06	16:10	Open unmanned systems icd draft v2-2 (aroc approved).docx Copy contents to clipboard Navigate to pastebin.com Paste contents and submit (Submit does not work on IE8)
UC3.S1	16:23	16:39	Connect USB hard drive to workstation Copy FirewallSource.zip to clipboard Disconnect USB Harddrive

Scenario	Start Time	Finish Time	Steps
UC4.S3	16:40	16:52	Open command line Navigate to desktop and make a folder named files copy z:\FirewallSource.zip to c:\Users\tgreen\Desktop\files Issue ftp –help Issue ftp martincrawford.net Transfer file and quit FTP session
UC3.S2	17:20	17:28	Mount workstation3\users\lscarlet to X Copy NewHire, Payroll, SocialSecurityNumber to C:\ Mount KernSafe virtual CD-RW Create CD named PayrollData Burn files using Windows functionality
UC4.S4	17:52	17:53	Open command prompt Connect to existing mapped network drive X: List directory contents Delete each file individually
UC6.S3	18:05	18:08	Open Outlook and access UAV Specs email Open Internet Explorer Copy contents of email Navigate to pastebin.com Paste email contents
UC4.S5	18:34	18:36	Create local administrator account Mallory Logoff and login as Mallory Connect USB hard drive Copy Payroll.xlsx to removable drive Disconnect removable drive
UC1.S3	00:02	00:10	Access computer at suspicious time Map \\MS01\Organization to local drive Copy Logger.cpp, PacketInspection.cpp, AutoUpdate.cpp, VM Configuration.xlsx, and Passwords.xlsx to the Desktop Print all documents to local printer
UC5.S1	00:12	00:21	Connect to workstation1 from personal computer via RDP Copy and paste FirewallSource.zip to C drive on home computer Copy JointStrikeFighter.docx to C drive on home computer Close RDP session
UC5.S2	00:22	00:31	Connect to MS01 from personal computer via RDP Mount workstation1\Users\tgreen to MS01 network drive Copy DarkCometRAT.exe from personal computer to MS01 Copy executable from MS01 to mounted network drive (workstation1) Close RDP session

*Denotes this set of actions is not a separate scenario. Listed times are DomU times.
**Denotes CMAT-V crashed during the execution of the scenario and was restarted.

Appendix D: Normal User Script (Workstations 1-3) Modified from [67]

Start	Finish	Host	Description
11:42	1145	w7_2	Performance review for mr. green
1145	1148	w7_2	email review for green
1148		w7_1	Read email
1148	1152	w7_3	Read email
		w7_2	map network drive (mustard to Z)
1151		w7_3	open ie
1152	1154	w7_3	google dog pics
Workstation times changed to pre-DST time (Now +1 of Dom0 time). Servers did not change.			
1254	1254	w7_3	copy dog pic to clipboard
1255	1256	w7_3	email mustard dog pic
1256	1256	w7_3	Close IE, close explorer
1257	1259	w7_1	Disable AV
1257	13:01	w7_2	Read dog picture email
1301	1308	w7_3	Create password protected file(newhire.docx)
1303	1303	w7_3	Map human resources network drive to Z
1312	1312	w7_2	Print green 2012 performance eval to lissard (Green-2012.docx)
1321	1334	w7_3	Navigate to www.reddit.com/r/funny
1322	1334	w7_3	compose email to col mustard with funnypic.jpg on clipboard and pasted to email
1332	1332	w7_2	Games do not work (no DirectX support)
1334	1335	w7_3	Reply to girls night out
1338	1340	w7_1	email col m ustard for raise
1339	1342	w7_2	Emails the 3 women for their evaluations
1342	1350	w7_2	reads mr. green's eval response
1342	1350	w7_3	reads ccol. Mustards evaluation email
1452	1254	w7_3	OK' email from scarlet to white
15:04	15:12	w7_2	Bath Salts search + images
15:09	15:12	w7_2	Add bath salts to favorites ('Figero')
15:12	15:12	w7_2	Search own pc for 'Bath Salts'
15:12	15:16	w7_3	Searach google for swinger, swingers, swinger pin, swinger bumper sticker
15:16	15:22	w7_3	Open swingerpin3 in image viewer
15:17	15:17	w7_1	direct navigate to ameritrade.com
15:17	15:17	w7_1	bookmark ameritrade
15:17	15:21	w7_1	google search for ameritrade
15:21	15:25	w7_1	search for ferrari

Start	Finish	Host	Description
15:27	15:28	w7_1	Email to peacock, scarlet, white
15:31	15:37	w7_3	Read 0k email
15:35	15:37	w7_1	Map network drive to mustard's performance evals (Y drive)
15:35	15:37	w7_1	Copy all evals to desktop and open White-2011.docx
15:43	15:46	w7_2	direct navigate to pga.com, search for golf, golf course, pga, pga tour
15:46	15:46	w7_2	Reply to green's raise email
15:48	15:48	w7_1	Reads mustard's email
16:09	16:09	w7_1	Black hat requist to mustard
16:09	16:14	w7_2	Mustard reply blackhat
16:15	16:15	w7_1	Email to Ms. White for tech specs on UAV
16:15	16:26	w7_1	read mustards email
16:16	16:16	w7_3	Swinger email to peacock
16:44	16:46	w7_1	Green drinks email. Included ferrari picture
16:46	16:46	w7_3	Scarlet reply to green
16:51		w7_3	Scarlet web browsing(reddit, facebook, gmail, youtube, pandora, amazon,chase)
16:58	16:59	w7_3	print document from chase page
17:08	17:18	w7_3	email reply for mr. plum. His SSN is on the clipboard, 534-23-1235
17:10	17:10	w7_2	Copy Crawford-2011.docx from network share to Desktop
17:21	17:21	w7_3	copy newHire.docx, payroll.xsls, socialSecurityNumbers.docx to My Documents
17:44		w7_2	Mustard web browsing(reddit, facebook, gmail, youtube, pandora, amazon,chase)
17:44	17:44	w7_2	Search for Crawford in mustard network share
17:44	17:44	w7_2	Open Crawford-2011.docx
17:57	17:58	w7_3	Emails mustard about missing files on the system
17:55	17:55	w7_3	"the expected revenue for 2012..." text on clipboard
18:03	18:03	w7_2	Forward scarlets email to security engineer
18:04	18:04	w7_2	lsPassword!123 text on clipboard
18:09	18:12	w7_3	Copy and paste swingers.txt contents from notepad to outlook
18:23		w7_3	Copy swingers.txt to swingers - Copy.txt. Rename to Swingers May Invite. Copy from Desktop to ...\lscarlet\
18:27		w7_2	Print Files Missing email from Scarlet to Lissard printer
18:28		w7_3	Print swingers.txt, Swingers May Invite.txt

Appendix E: Normal User Script (Workstations 4-5) Modified from [67]

Start	Host	Description
11:38	7_5	Opens Personal Information File in share drive. Open Payroll document. Excel document
11:43	7_5	Opens Internet Exporer. Googles Dogs Pictures.
11:46	7_5	Downloads Doggy Picture From Internet
11:48	7_5	Read Email, Microsoft Outlook
11:51	7_5	Close Outlook, Excell, and photo viewer
11:53		Googled "Divorce in Ohio"
Workstation times changed to pre-DST time (Now +1 of Dom0 time). Servers did not change.		
12:57	7_5	Down Divorce Documents for Court PDF file "otf.pdf"
12:59	7_4	Google Search "Cage fighting"
13:03	7_5	Opened Calc.exe
13:03	7_4	video results in search need flash installed to watch video. Installed flash from adobe.com and installed flash
13:06	7_5	Opened sticky notes software
13:10	CMAT	Memory Capture Seg Fault, Had to restart
13:12	7_4	Logged Off, 1 min later Logged on
13:14	7_4	Cage Fighting google search
13:15	7_4	Google Search "Octagon"
13:17	7_4	Google search "Cage Fighting Schedule dc"
13:20	7_5	Google search "Divorce Law"
13:20		Seg Fault Restarted CMAT
13:27	7_5	Opened outlook email
13:29	7_4	Opened outlook
13:32	7_4	Sent Email to Scarlet and Peacock
14:54	7_4	Google Search "Cage fighting schedule dc"
14:59	7_4	Downloaded Google chrome. IE keeps crashing
15:01	7_4	Started Using Chrome. Google Search "Cage fighting women dc"
15:04	7_4	Youtube Search "Cage Fighting Women"
15:05	7_4	Google Search "Cock Fighting"
15:05	7_4	Google Search"Cock Fighting DC"
15:08	7_4	Google Search "Cock Fighting Schedule"
15:15	7_4	Email Reply To Scarlet
15:18	7_5	Opened Paint. Made a Paint document and altered puppy picture that was saved on desktop
15:20	7_5	Opened Calculator

Start	Host	Description
15:21	7_5	Played Kalimba music (Under Sample Music), Could not play because not sound installed
15:23	7_5	Opened MS Excel
15:24	7_5	Opened MS power Point
15:24	7_5	Opened CMD window
15:25	7_5	Opened MS Words
15:27	7_5	dir command on CMD window
15:28	7_5	saved MS Excel named "numtest"
15:34	7_4	Sent Email response to Mr Green
15:36	7_5	Downloaded Chrome and installed chrome
15:38	7_5	Downloaded Firefox and installed firefox
15:41	7_5	opened firefox, chrome, and firefox
15:42	7_5	Using firefox went to ESPN.com
15:43	7_5	Using chrome went to facebook.com
15:45	7_5	Using IE went to reddit.com
16:03	7_4	Network Drive Opened "Joint Strike Fighter Program" MS WORD and printed the document. Printed it twice
16:14	7_4	Went to Pandora website and listened to internet radio. "Britney Spears Radio Stationed". Site not letting music play.
16:21	7_5	Open remote desktop shell to ms01
16:25	7_5	Opend classified project picture from remote desktop
16:27	7_4	Copied "Sleep Away" from Sample Music to Desktop using CTR C CTRV command
16:29	7_4	Copied "Sleep Away" from desktop to Document Library using CTR C CTR V
16:42	7_4	Created shortcut of shared network to desktop
16:43	7_4	Copied File "JointStrikeFighter" Work document from shared folder "Projects" to Desktop, Using Drag and Drop
16:48	7_4	Went to YouTube using Chrome
16:49	7_4	Went to facebook using another tab in Chrome
16:50	7_4	Went to Gmail.com using another tab in chrome
16:50	7_4	Went to reddit.com using another tab in chrome
16:52	7_4	Opened Notepad
16:54	7_4	Opened Calc.exe
16:56	7_5	Logged from remote shell
16:57	7_5	Went to facebook.com using another tab in IE
16:58	7_5	Went to Netflix.com using another tab in IE
16:59	7_5	Went to HULU using another tab in IE

Start	Host	Description
17:04	7_5	Email Response Sent to Scarlet
17:07	7_4	Email Response to MR GREEN
17:10	7_5	Went to ESPN.COM using another tab in IE
17:54	7_4	Created Word Doc for Fighter Design proposal in Documents Folder
17:55	7_4	Copy "draft copy of fighter design" from Document Folder to Desktop
18:05	7_4	cut and paste some text from "JointStrikeFighter" Word Document on desktop to "draft copy of fighter design" Word document on desktop using right click mouse. Saved Document
18:10	7_4	Copy "draft copy of fighter design" from desktop to network folder "Projects" using mouse cut and paste
18:12	7_5	closed all tabs of ID running
18:13	7_5	Closed all windows
18:24	7_4	Created folder in C:\ called "White_Folder"
18:25	7_4	Copied and Pasted 2 files using Highlight and right click copy and paste. Desktop to another folder in "C:\White_Folder"
18:28	7_5	Down loaded pdf.reader and installed it
18:37	7_5	Opened Divorce Document on Desktop and started filling it out
18:38	7_5	Printed the divorce document
18:45	7_5	View Doggy Picture on Windows Photo Viewer

Appendix F: ACE Hackfest Attack Log

The ACE Hackfest log data is very limited. Only a handful of attacks were recorded during the exercise. Additionally, none of the normal user behavior, such as email or document editing, was documented.

Src IP	Dest IP	Time	Malicious Activity
10.1.30.100	10.1.30.11	10:45	Metasploit DCOM Exploit w/ Reverse Shell
Unknown	Unknown	10:00-10:30	Mailbomb attack to BSOD email addresses (2000 emails sent)
10.1.80.212	10.1.30.10	09:30	Mailbomb postmaster@bsod.ace. 60 emails every 3 minutes w/ 25 MB attachments
10.1.30.11	10.1.80.212	13:00-13:10	LOIC
10.1.30.203	10.1.80.11	08:12	Nessus Internal Scan
10.1.30.203	10.1.80.0\24	08:15	Nmap Intense Scan
10.1.30.203	10.1.80.0\24	08:20	Hail Mary by Port
10.1.30.203	10.1.80.12	08:24	Nessus Internal Scan
10.1.30.203	10.1.80.0\24	08:35	Nmap Intense Scan
10.1.30.203	10.1.80.12	08:45	FTP Check Exploits
10.1.30.203	10.1.80.12	08:46	IIS check Exploits
10.1.30.203	10.1.80.112	08:48	FTP Check Exploits
10.1.30.203	10.1.80.10	08:50	SMTP Check Exploits
10.1.30.203	10.1.80.11	09:00	Nmap Intense Scan
10.1.80.71	10.1.30.11	09:12	Metasploit reverse shell 445
10.1.80.71	10.1.30.12	14:21	Unknown
10.1.80.71	10.1.30.11	13:39	Meterpreter Script
10.1.80.62	10.1.30.11	08:30	Pass-the-hash reverse TCP Shell

```
# BrowserHistory.py
# Martin Crawford - March 2012 - Master's Thesis - AFIT
#
# Attempts to extract browsing history from memory captures.
#      This method produces more false positive results, but will
guarantee to capture all instances
#
#
import os
import fnmatch
import array
import re
import string
import sys
DEBUG = 0

microsoftstr = "\x6d\x69\x63\x72\x6f\x73\x6f\x66\x74" # "microsoft"

httpsspaced =
"\x68\x00\x74\x00\x74\x00\x70\x00\x73\x00\x3A\x00\x2F\x00\x2F\x00"
                          # h.t.t.p.s.:./../.
httpspaced = "\x68\x00\x74\x00\x74\x00\x70\x00\x3A\x00\x2F\x00\x2F\x00"
                                      # h.t.t.p.:./../.
faviconspaced =
"\x66\x00\x61\x00\x76\x00\x69\x00\x63\x00\x6f\x00\x6e\x00\x2e\x00\x69\x
00\x63\x00\x6f"   # f.a.v.i.c.o.n...i.c.o
htmspaced =
"\x00\x63\x00\x6F\x00\x6D\x00\x5B\x00\x31\x00\x5D\x00\x2E\x00\x68\x00\x
74\x00\x6D"            # .c.o.m.[.1.]...h.t.m

# Pass in the folder to start the scanning.
rootdir = sys.argv[1]

for root, subFolders, files in os.walk(rootdir):
      for file in files:
            if fnmatch.fnmatch(file, '*.dmp'):
                  if DEBUG:
                        print "Found .dmp file"
                  filePath = os.path.join(root, file)            #
Full path including filename
                  if DEBUG:
                        print "\n\tOpening File" +    filePath
                  filename = "_history"
                  infile = open(filePath, "rb")
                  dmp = infile.read()
                  infile.close()
                  listindex = []
                  listindex2 = []

                  # httpspaced
                  offset = 0
                  scanstr = httpspaced
                  i = dmp.find(scanstr, offset)
```

```
                while i >= 0:
                        listindex.append(i)
                        i = dmp.find(scanstr, i + 1)

                # htmspaced
                offset = 0
                scanstr = htmspaced
                i = dmp.find(scanstr, offset)
                while i >= 0:
                        listindex2.append(i)
                        i = dmp.find(scanstr, i + 1)
                if DEBUG:
                        print listindex

                # Write results to file
                logdir = root + "\History\\"
                if DEBUG:
                        print "Checking file path: " + logdir
                if not os.path.exists(logdir):
                        os.makedirs(logdir)


                historyjobfile = open(logdir +file[:17] + filename +
".log", "w")
                cleanhist = []

                for histitems in listindex:
                        histinfo = dmp[histitems:histitems+80]
                        histinfo = histinfo.replace("\0", "")      #
Remove null padding
                        histinfo = histinfo.split("/")

                        # Don't die if histinfo[2] doesnt exist
                        try:
                                # there are hundreds of
support.microsoft.com urls in Windows. Probably can't perform any
malicious actions there anyway.
                                if "microsoft" not in histinfo[2]:
                                        # Only print if it does not already
exist
                                        if histinfo[2] not in cleanhist:
                                                # Make sure it is an actual
URL and not clobbered by memory
                                                if all(c in string.printable
for c in histinfo[2]):
                                                        # URL needs to have a
letter + .TLD
                                                        if
(histinfo[2].__len__() > 5):

     cleanhist.append(histinfo[2])

     historyjobfile.write("Possible Browser History Found at: " +
hex(histitems) + "\n")
```

```
        historyjobfile.write(histinfo[2])

        historyjobfile.write("\n\n")
                        except IndexError, e:
                                print e
                historyjobfile.close()
                # Clean up memory so it doesn't crash.
                # Python doesn't clean up this variable before trying
to open the next .dmp
                dmp = "\0"
```

Appendix H: File Download Extraction Script

```python
# Downloads.py
# Martin Crawford - March 2012 - Master's Thesis - AFIT
#
# Finds possible file downloads
#
#
import os
import fnmatch
import array
import re
import string
import gc
import time
import sys
DEBUG = 1

microsoftstr = "\x6d\x69\x63\x72\x6f\x73\x6f\x66\x74"
downloadsstr =
"\x00\x3A\x00\x5A\x00\x6F\x00\x6E\x00\x65\x00\x2E\x00\x49\x00\x64\x00\x
65\x00\x6E\x00\x74\x00\x69\x00\x66\x00\x69\x00\x65\x00\x72\x00"


# Pass in the folder to start the scanning.
rootdir = sys.argv[1]


for root, subFolders, files in os.walk(rootdir):
      for file in files:
            if fnmatch.fnmatch(file, '*.dmp'):
                  if DEBUG:
                        print "Found .dmp file"
                  filePath = os.path.join(root, file)              #
Full path including filename
                  if DEBUG:
                        print "\n\tOpening File" +    filePath
                  filename = "_downloads"
                  infile = open(filePath, "rb")
                  dmp = infile.read()
                  infile.close()
                  listindex = []
                  offset = 0
                  i = dmp.find(downloadsstr, offset)
                  while i >= 0:
                        listindex.append(i)
                        i = dmp.find(downloadsstr, i + 1)
                  if DEBUG:
                        print listindex

                  # Write results to file
                  logdir = root + "\Downloads\\"
                  if DEBUG:
                        print "Checking file path: " + logdir
                  if not os.path.exists(logdir):
```

```python
                os.makedirs(logdir)

            cleanhist = []

            downloadsfile = open(logdir +file[:17] + filename +
".log", "w")
            for downloaditems in listindex:
                try:
                    downinfo = dmp[downloaditems-
128:downloaditems]
                    downinfo = downinfo.replace("\0", "")
    # Remove null padding

                    cleanhist.append(downinfo)
                    downloadsfile.write("Possible File
Download Found at: " + hex(downloaditems) + "\n")
                    downloadsfile.write(downinfo)
                    downloadsfile.write("\n\n")
                except IndexError, e:
                    print e

            downloadsfile.close()
            # Clean up memory so it doesn't crash.
            # Python doesn't clean up this variable before trying
to open the next .dmp
            dmp = "\0"
```

Appendix I: Email Extraction Script

```python
# Email.py
# Martin Crawford - March 2012 - Master's Thesis - AFIT
#
# Attempts to extract email contents from a memory capture.
#      Script is alpha quality. Seems to capture all HTML based emails,
but lots of extra noise.
#
#
import os
import fnmatch
import array
import sys
DEBUG = 1

emailstr =
"\x3C\x68\x74\x6D\x6C\x20\x78\x6D\x6C\x6E\x73\x3A\x76\x3D\x22\x75\x72\x
6E\x3A\x73\x63\x68\x65\x6D\x61\x73\x2D\x6D\x69\x63\x72\x6F\x73\x6F\x66\
x74\x2D\x63\x6F\x6D\x3A\x76\x6D\x6C\x22\x20\x78\x6D\x6C\x6E\x73\x3A\x6F
\x3D\x22\x75\x72\x6E\x3A\x73\x63\x68\x65\x6D\x61\x73\x2D\x6D\x69\x63\x7
2\x6F\x73\x6F\x66\x74\x2D\x63\x6F\x6D\x3A\x6F\x66\x66\x69\x63\x65\x3A\x
6F\x66\x66\x69\x63\x65\x22"

# Pass in the folder to start the scanning.
rootdir = sys.argv[1]


for root, subFolders, files in os.walk(rootdir):
      for file in files:
            if fnmatch.fnmatch(file, '*.dmp'):
                  if DEBUG:
                        print "Found .dmp file"
                  filePath = os.path.join(root, file)                #
Full path including filename
                  if DEBUG:
                        print "\n\tOpening File: " +  filePath
                  filename = "_email"
                  infile = open(filePath, "rb")
                  dmp = infile.read()
                  infile.close()
                  listindex = []
                  offset = 0
                  i = dmp.find(emailstr, offset)
                  while i >= 0:
                        listindex.append(i)
                        i = dmp.find(emailstr, i + 1)
                  if DEBUG:
                        print listindex

                  # Write results to file
                  logdir = root + "\Email\\"
                  if DEBUG:
                        print "Checking file path: " + logdir
                  if not os.path.exists(logdir):
```

186

```
                    os.makedirs(logdir)


                emailjobfile = open(logdir +file[:17] + filename +
".log", "w")
                for privatejob in listindex:
                    jobinfo = dmp[privatejob:privatejob+3327]
    #0xCFF bytes
                    emailjobfile.write("Possible Email found at: "
+ hex(privatejob) + "\n")
                    jobinfo = jobinfo.replace("\0", "")
                    emailjobfile.write(jobinfo)
                    emailjobfile.write("\n\n")
                emailjobfile.close()
                # Clean up memory so it doesn't crash.
                # Python doesn't clean up this variable before trying
to open the next .dmp
                dmp = "\0"
```

Appendix J: Print Job Extraction Script

```
# PrintJobExtract.py
# Martin Crawford - March 2012 - Master's Thesis - AFIT
#
# Finds possible references to print jobs
#     This method produces more false positive results, but will
guarantee* to capture all instances
#
# Big thanks to http://code.activestate.com/recipes/499314-find-all-
indices-of-a-substring-in-a-given-string/
#     for the sample string.find code. Byte by byte parsing is
unbelievably slow.
#
import os
import fnmatch
import array
import sys
DEBUG = 0
OFFICEONLY = 1

# Pass in the folder to start the scanning.
rootdir = sys.argv[1]


for root, subFolders, files in os.walk(rootdir):
      for file in files:
            if fnmatch.fnmatch(file, '*.dmp'):
                  if DEBUG:
                        print "Found .dmp file"
                  if OFFICEONLY:    # Only MS Office products
                        printstr =
"\x4E\x00\x54\x00\x20\x00\x45\x00\x4D\x00\x46\x00\x20\x00\x31\x00\x2E\x
00\x30\x00\x30\x00\x38\x00\x00\x00\x4D"
                        filename = "_OfficePrints"
                  else: # Produces more false positives, but also
captures more print jobs
                        printstr =
"\x4E\x00\x54\x00\x20\x00\x45\x00\x4D\x00\x46\x00\x20\x00\x31\x00\x2E\x
00\x30\x00\x30\x00\x38\x00\x00\x00"
                        filename = "_AllPrints"
                  filePath = os.path.join(root, file)            #
Full path including filename
                  if DEBUG:
                        print "\n\tOpening File" +    filePath


                  infile = open(filePath, "rb")
                  dmp = infile.read()
                  infile.close()
                  listindex = []
                  offset = 0
                  i = dmp.find(printstr, offset)
                  while i >= 0:
                        listindex.append(i)
```

```
                    i = dmp.find(printstr, i + 1)
              if DEBUG:
                    print listindex

              # Write results to file
              logdir = root + "\PrintJob\\"
              if DEBUG:
                    print "Checking file path: " + logdir
              if not os.path.exists(logdir):
                    os.makedirs(logdir)


              printjobfile = open(logdir +file[:17] + filename +
".log", "w")
              for printjob in listindex:
                    jobinfo = dmp[printjob:printjob+255]
                    printjobfile.write("Possible Print Job Found
at: " + hex(printjob) + "\n")
                    jobinfo = jobinfo.replace("\0", "")
                    printjobfile.write(jobinfo)
                    printjobfile.write("\n\n")
              printjobfile.close()
              # Clean up memory so it doesn't crash.
              # Python doesn't clean up this variable before trying
to open the next .dmp
              dmp = "\0"
```

Appendix K: InPrivate Browsing History Extraction Script

```
# PrivateBrowsing.py
# Martin Crawford - March 2012 - Master's Thesis - AFIT
#
# Extracts Internet Explorer InPrivate browsing history from a memory
capture
#
import os
import fnmatch
import array
import sys
DEBUG = 0

privatestr =
"\x49\x00\x6E\x00\x74\x00\x65\x00\x72\x00\x6E\x00\x65\x00\x74\x00\x20\x
00\x45\x00\x78\x00\x70\x00\x6C\x00\x6F\x00\x72\x00\x65\x00\x72\x00\x20\
x00\x2D\x00\x20\x00\x5B\x00\x49\x00\x6E\x00\x50\x00\x72\x00\x69\x00\x76
\x00\x61\x00\x74\x00\x65\x00\x5D"

# Pass in the folder to start the scanning.
rootdir = sys.argv[1]


for root, subFolders, files in os.walk(rootdir):
      for file in files:
            if fnmatch.fnmatch(file, '*.dmp'):
                  if DEBUG:
                        print "Found .dmp file"
                  filePath = os.path.join(root, file)              #
Full path including filename
                  if DEBUG:
                        print "\n\tOpening File" +    filePath
                  filename = "_inprivate"
                  infile = open(filePath, "rb")
                  dmp = infile.read()
                  infile.close()
                  listindex = []
                  offset = 0
                  i = dmp.find(privatestr, offset)
                  while i >= 0:
                        listindex.append(i)
                        i = dmp.find(privatestr, i + 1)
                  if DEBUG:
                        print listindex

                  # Write results to file
                  logdir = root + "\InPrivate\\"
                  if DEBUG:
                        print "Checking file path: " + logdir
                  if not os.path.exists(logdir):
                        os.makedirs(logdir)
```

```
                    privatejobfile = open(logdir +file[:17] + filename +
".log", "w")
                    for privatejob in listindex:
                          jobinfo = dmp[privatejob-128:privatejob]
                          privatejobfile.write("InPrivate browsing found
at: " + hex(privatejob) + "\n")
                          jobinfo = jobinfo.replace("\0", "")
                          privatejobfile.write(jobinfo)
                          privatejobfile.write("\n\n")
                    privatejobfile.close()
                    # Clean up memory so it doesn't crash.
                    # Python doesn't clean up this variable before trying
to open the next .dmp
                    dmp = "\0"
```

Bibliography

[1]   T. Garfinkel and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection," in *Proc. Network and Distributed Systems Security Symposium*, 2003.

[2]   A. Capalik, "Next-Generation Honeynet Technology with Real-Time Forensics for US Defense," in *Military Communications Conference, 2007. MILCOM 2007. IEEE*, 2007, pp. 2.

[3]   S. Chamotra, J. Bhatia, R. Kamal, and A. Ramani, "Deployment of a low interaction honeypot in an organizational private network," in *Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on*, 2011, pp. 130-131.

[4]   J. Hallahan, "Countering Insider Threats - Handling Insider Threats Using Dynamic, Run-Time Forensics," AFRL-RI-RS-TR-2007-213, Rome, NY, 2009.

[5]   "Insider Threats," Defense Security Service, Available: www.dss.mil/documents/ci/Insider-Threats.pdf, pp. 1-2.

[6]   "Examining Insider Threat Risk at the U.S. Citizenship and Immigration Services (Redacted)," Department of Homeland Security, Office of Inspector General, OIG-11-33, Washington, DC, 2011, pp. 2

[7]   W. Baker et al.,"2011 Data Breach Investigations Report," Verizon RISK Team, Available: http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2011_en_xg.pdf, pp. 1-72 2011.

[8]   S. L. Pfleeger, J. B. Predd, J. Hunker, and C. Bulford, "Insiders Behaving Badly: Addressing Bad Actors and Their Actions," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 169-179, Mar. 2010.

[9]   V. N. L. Franqueira, A. van Cleeff, P. Van Eck, and R. Wieringa, "External Insider Threat: a Real Security Challenge in Enterprise Value Webs," in *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, 2010, pp. 446–453.

[10] D. Cappelli, A. Moore, R. Trzeciak, and T. J. Shimeall, "Common Sense Guide to Prevention and Detection of Insider Threats (version 3.1)," CERT/Software Engineering Institute, Available: www.cert.org/archive/pdf/CSG-V3.pdf

[11] M. Maybury, "Detecting Malicious Insiders in Military Networks," *Military Communications Conference, 2006. MILCOM 2006*, Washington, DC, pp. 1-7, 2006.

[12] B. Gabrielson et al., "The Insider Threat to Information Systems: A State-of-the-Art Report," Contract SPO700-98-D-4002. Herndon VA: Information Assurance Technology Analysis Center (IATAC), October, 2008

[13] E. D. Shaw, K. G. Ruby and J. M. Post. "The Insider Threat to Information Systems: The Psychology of the Dangerous Insider," Security Awareness Bulletin, 2-98, Department of Defense Security Institute, Richmond, Virginia. September 1998.

[14] M.R. Randazzo et al., "Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector", Carnegie Mellon Univ., Software Eng. Inst., 2004; Available: www.cert.org/archive/pdf/bankfin040820.pdf

[15] E. Kowalski et al., "Insider Threat Study: Illicit Cyber Activity in the Government Sector," U.S. Secret Service and Carnegie Mellon Univ., Software Eng. Inst., Available: www.cert.org/archive/pdf/insiderthreat_gov2008.pdf, 2008.

[16] E. Kowalski et al., "Insider Threat Study: Illicit Cyber Activity in the Information Technology and Telecommunications Sector," U.S. Secret Service and Carnegie Mellon Univ., Software Eng. Inst., Available: www.cert.org/archive/pdf/insiderthreat_it2008.pdf, 2008.

[17] M. Keeney et al., "Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors", U.S. Secret Service and Carnegie Mellon Univ., Software Eng. Inst., Available: www.cert.org/archive/pdf/insidercross051105.pdf, 2005.

[18] "The National Counterintelligence Strategy of the United States of America", 2007, Available: www.dni.gov/reports/NCIX_Strategy_2007.pdf [Sep. 15, 2011]

[19] M. Balakgie, "Computer Security: Cyber Attacks – A War without Borders," Testimony to House Subcommittee on Government Management, Information and Technology, Jul. 26, 2000, Available: www.iwar.org.uk/cip/resources/congress/000726mb.htm [Nov. 11, 2011]

[20] "Data Loss Risks During Downsizing: As Employees Exit, so does Corporate Data", Ponemon Institute/Symantec Corp., Feb 23, 2009 Available: www.ponemon.org/local/upload/fckjail/generalcontent/18/file/Data Loss Risks During Downsizing FINAL 1.pdf [Sep. 15, 2011]

[21] H. Lipson, "Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues". CERT Coordination Center, Special Report CMU/SEI-2002-SR-009, 2002.

[22] E. Messmer, "Sys Admin Gone Rogue is Biggest Insider Threat," *Network World*, vol. 27, no. 18, pp. 11-13, Sep 27 2010.

[23] R. McMillan, "San Francisco IT Admin Locks Up City Network," *Computer World*, pp. 6, Jul 21 2008.

[24] F. Hayes, "Crazy Time", *Computer World,* pp. 44, Feb 9 2009.

[25] J. Vijayan, "City Missed Steps to Avoid Network Lockout," *Computer World,* pp. 16, Jul 28 2008.

[26] J. Barnes, and N. Hodge, "Military Probe Again Targets Manning," *The Wall Street Journal*, Jul 28 2010, Available: online.wsj.com/article/SB10001424052748703292704575393591013859452.html [Dec 2 2011]

[27] D. Leigh, "How 250,000 US Embassy Cables Were Leaked," *The Guardian*, Nov 28 2010, Available: www.guardian.co.uk/world/2010/nov/28/how-us-embassy-cables-leaked [Dec 2 2011]

[28] P. Earley, "'Treason?' He Repeats the Word out Loud as if He is Shocked by it," *U.S. News & World Report*, vol. 122, no. 6, pp. 29, 1997.

[29] J. Fialka, "Ames Gets Life in Prison for Spying as Part of Plea Deal to Protect Wife," *Wall Street Journal*, New York, NY, 1994, pp. A16.

[30] W. F. Buckley, "Was Ames a Mortal Blow?," *National Review*, vol. 47, no. 14, 1995, pp. 70-71.

[31] "USA v. Robert Philip Hanssen: Affidavit in Support of Criminal Complaint, Arrest Warrant, and Search Warrants," US District Court for the Eastern District of

Virginia, 2001.

[32] B. D. Payne, M. Carbone, and W. Lee. "Secure and Flexible Monitoring of Virtual Machines." *Proceedings of the Annual Computer Security Applications Conference*, 2007.

[33] P. Hoffman, K. Scarfone, and M. Souppaya. "Guide to Security for Full Virtualization Technologies," National Institute of Standards and Technology (NIST), 2011, Available: csrc.nist.gov/publications/nistpubs/800-125/SP800-125-final.pdf [Nov 23 2011]

[34] P. Hoffman, K. Scarfone, and M. Souppaya. "Guide to Enterprise Telework and Remote Access Security," National Institute of Standards and Technology (NIST), 2009, Available: csrc.nist.gov/publications/nistpubs/800-46-rev1/sp800-46r1.pdf [Nov 23 2011]

[35] T. Ables, P. Dhawan, and B. Chandrasekaran, "An Overview of Xen Virtualization," *Dell Power Solutions*, 2005, Available: www.dell.com/downloads/global/power/ps3q05-20050191-Abels.pdf [Nov 23 2011]

[36] R. F. Mills, et al., "A Scenario-Based Approach to Mitigating the Insider Threat," *ISAA Journal*, vol. 9 no. 5, 2011 pp. 12-19.

[37] T. Garfinkel and M. Rosenblum. "A Virtual Machine Introspection Based Architecture for Intrusion Detection," In *Proceedings of the 2003 Network and Distributed System Security Symposium (NDSS)*, 2003.

[38] B. Dolan-Gavitt, et al., "Virtuoso: Narrowing the Semantic Gap in Virtual Machine Introspection," In *Proc. IEEE Symposium on Security and Privacy*, 2011.

[39] B. Hay, M. Bishop, and K. Nance, "Live Analysis: Progress and Challenges," *IEEE Security and Privacy*, vol. 7, Mar. 2009, pp. 30-37

[40] C. Davis, D. Cowen, and A. Philipp, *Hacking Exposed: Computer Forensics Secrets & Solutions*, 2nd ed. New York: McGraw-Hill, 2010, pp. 65, 458-459.

[41] S. Mrdovic, A. Huseinovic, and E. Zajko, "Combining Static and Live Digital Forensic Analysis in Virtual Environment," *XXII International Symposium on Information, Communication and Automation Technologies*, 2009, pp. 1-6.

[42] H. Carvey, *Windows Forensic Analysis DVD Toolkit*, 2nd ed. Burlington MA: Syngress Publishing, 2009 pp. 3-4, 158-251.

[43] B. Carrier, "Risk of Live Digital Forensic Analysis", *Communications of the ACM* vol. 49, no. 2, 2006, pp. 56-61.

[44] G. Hoglund. "Rootkits: Subverting the Windows Kernel," Addison Wesley, 2005.

[45] J. Okolica and G. Peterson, "Windows Operating Systems Agnostic Memory Analysis," in *Proceedings of the Digital Forensic Research Workshop Conference (DFRWS)*, 2010.

[46] J. Okolica and G. Peterson, "Extracting the Windows Clipboard from Physical Memory," *Digital Investigations Journal*, 2011 pp. 118-124.

[47] D. Dodge, "Cyber-Situational Awareness Using Live Hypervisor-Based Virtual Machine Introspection", M.S. thesis, ENG, AFIT, Wright-Patterson AFB, OH, 2010.

[48] S. Hansman and R. Hunt, "A Taxonomy of Network and Computer Attacks Computers and Security," *Elsevier*, vol 24, no 1, 2005, pp. 31-43.

[49] C. Ellis, et al., "AVOIDIT: A Cyber Attack Taxonomy," Technical Report: CS-09-003, University of Memphis, 2009.

[50] J. D. Howard and T. A. Longstaff, "A Common Language for Computer Security Incidents," Technical Report SAND98-8667, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 1998.

[51] R. McGrew, R. Vaughn, "Experiences with Honeypot Systems: Development, Deployment, and Analysis", in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences* 2006.

[52] L. Spitzner, "Honeypots: Catching the Insider Threat", *Annual Computer Security Applications Conference* December 2003.

[53] S. Chamotra, et al., "Deployment of a Low Interaction Honeypot in an Organizational Private Network", *IEEE International Conference on Emerging Trends in Networks and Computer Communications* (ETNCC '11), pp. 130-135, 2011.

[54] L. Spitzner. (2003 Jul. 17) *Honeytokens: The Other Honeypot.* [Online] Available: http://www.securityfocus.com/infocus/1713

[55] Center for Cyberspace Research, *Advanced Cyber Education* [Online]. Available: http://www.afit.edu/ccr/ace/docs/Prospective_ACE_Student_Brochure.pdf

[56] Center for Cyberspace Research, *General Course Information* [Online]. Available: http://www.afit.edu/ccr/ace/docs/general_course_information.pdf

[57] K. Mandia, C. Prosise, and M. Pepe, *Incident Response & Computer Forensics*, 2nd ed. New York, NY: McGraw-Hill, pp. 312-314.

[58] Microsoft Corporation, *Windows Registry Information for Advanced Users* [Online]. Available: http://support.microsoft.com/kb/256986 Feb, 2008, [Mar 5, 2012].

[59] N.Hirst, "The Implications of Virtual Machine Introspection for Digital Forensics on Nonquiescent Virtual Machines", M.S. thesis, Naval Postgraduate School, Monterey, CA, 2011.

[60] B. Hay and K. Nance, "Forensics examination of volatile system data using virtual introspection," SIGOPS Oper. Syst. Rev., vol. 42, no. 3, pp. 74–82, 2008

[61] QEMU. (2012, March) [Online]. http://wiki.qemu.org

[62] Nabble. (2012, March) Passing USB device to MS Windows guest. [Online]. http://xen.1045712.n5.nabble.com/Passing-USB-device-to-MS-Windows-guest-td2600029.html

[63] Xen. (2012, March) Xen USB Passthrough. [Online]. http://wiki.xen.org/wiki/Xen_USB_Passthrough

[64] Olivetalks. (2012, March) USB forwarding on Xen – it just does not work. [Online]. http://www.olivetalks.com/2008/02/03/usb-forwarding-on-xen-it-just-does-not-work/

[65] FabulaTech LLP. (2012, March) USB over Network. [Online]. http://www.usb-over-network.com/usb-over-network.html

[66] Citrix. (2012, March) Xen-Users Mailing List Archive – Can't use DVD Write in DomU. [Online]. http://old-list-archives.xen.org/archives/html/xen-users/2008-02/msg00527.html

[67] KernSafe (2012, March) TotalMounter. [Online].

http://www.kernsafe.com/product/totalmounter.aspx

[68] Millette, C. D., "*Air Force Officials Initiate Service-Wide Upgrade to Windows 7* [Online]. Available: http://www.af.mil/news/story.asp?id=123228739

[67] Swartzmiller, Maj., Simpson, Capt., Sievers, Capt., "Live Network Forensic Response", Final Project, CSCE 527, Air Force Institute of Technology, WPAFB, OH, Sep. 2, 2011, pp. 9-11.

[68] CERT, (2012, March) Data Exfiltration and Output Devices – An Overlooked Threat. [Online]. https://www.cert.org/blogs/insider_threat/2011/10/data_exfiltration_and_output_ devices_-_an_overlooked_threat.html

[69] Hewlett-Packard. (2012, March). HP Access Control Printing Solutions [Online]. http://h71028.www7.hp.com/enterprise/us/en/ipg/access-control-printing-solutions.html

[70] United States Air Force. (2012, April). Thumb drives/flash media still prohibited on Air Force Network [Online]. http://www.af.mil/news/story.asp?id=123192400

[71] Google Code. (2012, April). Volatility - Issue 148: Add Windows commandline history records plugin [Online]. https://code.google.com/p/volatility/issues/detail?id=148

[72] Microsoft. (2012, April). PsTools [Online]. http://technet.microsoft.com/en-us/sysinternals/bb896649

[73] Microsoft. (2012, January). Shell Clipboard Formats [Online]. http://msdn.microsoft.com/en-us/library/windows/desktop/bb776902%28v=vs.85%29.aspx

[74] Microsoft. (2012, January). STGMEDIUM structure [Online]. http://msdn.microsoft.com/en-us/library/windows/desktop/ms683812%28v=vs.85%29.aspx

[75] Microsoft. (2012, January). Handling Shell Data Transfer Scenarios [Online]. http://msdn.microsoft.com/en-us/library/windows/desktop/bb776904%28v=vs.85%29.aspx

[76] Microsoft. (2012, January). IDataObject Interface [Online]. http://msdn.microsoft.com/en-us/library/windows/desktop/ms688421%28v=vs.85%29.aspx

[77] W. Bai, "Development of a Methodology for Customizing Insider Threat Auditing on a Linux Operating System", M.S. thesis, ENG, AFIT, Wright-Patterson AFB, OH, 2010.

[78] T. Levoy, "Development of a Methodology for Customizing Insider Threat Auditing on a Microsoft Windows XP® Operating System", M.S. thesis, ENG, AFIT, Wright-Patterson AFB, OH, 2006.

[79] J. Butts, "Formal Mitigation Strategies for the Insider Threat: A Security Model and Risk Analysis Framework", M.S. thesis, ENG, AFIT, Wright-Patterson AFB, OH, 2006.

[80] M. Woolingham, "Detecting Insider Threats on a Cisco Router Using the Native Functionality of the Internetwork Operating System", M.S. thesis, ENG, AFIT, Wright-Patterson AFB, OH, 2011.

[81] J. Myers, "A Dynamically Configurable Log-Based Distributed Security Event

Detection Methodology Using Simple Event Correlator", M.S. thesis, ENV, AFIT, Wright-Patterson AFB, OH, 2010.

[82] National Security Agency. (2011, December). Defense in Depth [Online]. www.nsa.gov/ia/_files/support/defenseindepth.pdf

[83] Y. Liu, C. Corbett, K. Chiang, R. Archibald, B. Mukherjee, and D. Ghosal, 'SIDD: A framework for detecting sensitive data exfiltration by an insider attack', in *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, 2009, pp. 1–10.

[84] S. T. Jones, A. C. Arpaci-Dusseau, and R. H. Arpaci Dusseau. "Antfarm: Tracking processes in a virtual machine environment". In *Proceedings of the USENIX Annual Technical Conference*, June 2006

[85] Backbone Security. (2012, February). Backbone Security Announces Enhanced Insider Threat Detection Capability. [Online] http://www.backbonesecurity.com/InsiderThreatDetection.aspx

[86] N. Nguyen, P. Reiher, G.H. Kuenning, "Detecting insider threats by monitoring system call activity", *IEEE Information Assurance Workshop*, pp. 45-52, June 2003.

[87] Rayethon. (2012, March). Rayethon SureView. [Online] http://www.raytheon.com/capabilities/products/cybersecurity/insiderthreat/products/surview/index.html

[88] Symantec. (2012, January). Preventing and Detecting Insider Attacks Using IDS. [Online] http://www.symantec.com/connect/articles/preventing-and-detecting-insider-attacks-using-ids

[89] M. Bishop, *Computer Security: Art and Science*, 2003, Westford, MA: Addison Wesley Professional, 2003, pp. 4-12.

[90] K. Thompson, "Reflections on trusting trust", Comm. of ACM, vol. 27, no. 8, pp. 761-763, Aug. 1984

1. REPORT DATE (DD-MM-YYYY) 14-06-2012	2. REPORT TYPE Master's Thesis	3. DATES COVERED (From – To) Sep 2010 – Jun 2012

4. TITLE AND SUBTITLE Insider Threat Detection on the Windows Operating System using Virtual Machine Introspection	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Crawford, Martin H., Civ, USAF	5d. PROJECT NUMBER N/A
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GCO/ENG/12-15
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Security Agency Attn: Kelly Hughes 9800 Savage Road Suite 6767 Ft. Meade, MD 20755-6767 Commercial 410-854-6972 DSN 244-6972 k.hughes@radium.ncsc.mil	10. SPONSOR/MONITOR'S ACRONYM(S) NSA/CND R&T
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
Existing insider threat defensive technologies focus on monitoring network traffic or events generated by activities on a user's workstation. This research develops a methodology for signaling potentially malicious insider behavior using virtual machine introspection (VMI). VMI provides a novel means to detect potential malicious insiders because the introspection tools remain transparent and inaccessible to the guest and are extremely difficult to subvert. This research develops a four step methodology for development and validation of malicious insider threat alerting using VMI. Six core use cases are developed along with eighteen supporting scenarios. A malicious attacker taxonomy is used to decompose each scenario to aid identification of observables for monitoring for potentially malicious actions. The effectiveness of the identified observables is validated through the use of two data sets, one containing simulated normal and malicious insider user behavior and the second from a computer network operations exercise. Compiled Memory Analysis Tool – Virtual (CMAT-V) and Xen hypervisor capabilities are leveraged to perform VMI and insider threat detection. Results of the research show the developed methodology is effective in detecting all defined malicious insider scenarios used in this research on Windows guests.

15. SUBJECT TERMS
Insider Threat, Virtual Machine Introspection, Registry Forensics, Virtual Machine, Xen Hypervisor

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Gilbert L. Peterson, PhD (ENG)
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	UU	213	19b. TELEPHONE NUMBER (Include area code) (937) 255-6565, x 4281 (gilbert.peterson@afit.edu)